Semantic Web techniques to support interoperability in distributed network environments

Dave Braines*, Yannis Kalfoglou†, Paul Smart†, Jie Bao‡, Nigel Shadbolt† and James Hendler‡

*Emerging Technology Services, IBM United Kingdom Ltd, Hursley Park, Winchester, UK.
Email: dave_braines@uk.ibm.com
†School of Electronics and Computer Science, University of Southampton, UK.
Email: {y.kalfoglou,ps02v,nrs}@ecs.soton.ac.uk
‡Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, USA.
Email: {baojie,hendler}@cs.rpi.edu

***Abstract*—We explore two Semantic Web techniques arising from ITA research into semantic alignment and interoperability in distributed networks. The first is *POAF* (Portable Ontology Aligned Fragments) which addresses issues relating to the portability and usage of ontology alignments. *POAF* uses an ontology fragmentation strategy to achieve portability, and enables subsequent usage through a form of automated ontology modularization. The second technique, *SWEDER* (Sematic Wrapping of Existing Data sources with Embedded Rules), is grounded in the creation of lightweight ontologies to semantically wrap existing data sources, to facilitate rapid semantic integration through representational homogeneity. The semantic integration is achieved through the creation of *context ontologies* which define the integrations and provide a portable definition of the integration rules in the form of embedded SPARQL construct clauses. These two Semantic Web techniques address important practical issues relevant to the potential future adoption of ontologies in distributed network environments.**

I. Introduction

Research sponsored by the International Technology Alliance[1] (ITA) into semantic interoperability and alignment techniques has recently yielded two promising techniques each of which addresses a different, but important, set of challenges. One specific focus of this research has been on delivering a pragmatic semantic interoperability solution in an environment using distributed networks which are potentially unstable and are therefore considered unreliable. The specific operational context for our ITA research has been that of a coalition of military, non-government organisation (NGO) and local force alliance members working together in a distributed network environment. Wherever possible our ITA research is based on de-facto Semantic Web standards and existing technology components.

Both of the techniques discussed in this paper have been introduced and described in detail in previous publications [1], [2]. The specific purpose of this paper is to provide an overall description of each technique and bring both techniques together in a single example scenario to demonstrate their fundamental capabilities working together.

The first of these two techniques is *POAF* (Portable Ontology Aligned Fragments) [1] which addresses issues relating to the portability and subsequent usage of ontology alignments. *POAF* uses an ontology fragmentation strategy to achieve portability with subsequent usage supported through a form of automated ontology modularization.

The second technique is *SWEDER* (Semantic Wrapping of Existing Data sources with Embedded Rules) [2] and is grounded in the creation of lightweight ontologies to semantically wrap existing data sources in order to facilitate rapid semantic integration through representational homogeneity. One important form of semantic integration can be achieved through the creation of *context ontologies* which are built upon the *SWEDER* technique and which have the specific purpose of defining the integrations and providing a portable definition of the rules to achieve these integrations in the form of SPARQL[2] construct clauses.

Since this paper is primarily intended to describe the progress on these two specific techniques, the main content is found in sections II and III which give an overview of *POAF* and *SWEDER* respectively. These two sections consolidate materials from the previous publications [1], [2], introducing each of the techniques and giving an appropriate overview of the details. Section IV provides an example scenario demonstrating the benefits of both of these techniques and draws their usage together in a single context. Section V gives details of other work which is related to the *POAF* and *SWEDER* techniques, and the paper is concluded in section VI.

II. POAF - Portable Ontology Aligned Fragments

A key concern in our domain of interest 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 [3] but

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

[2]http://www.w3.org/TR/rdf-sparql-query/

coalition operations impose strict availability and access constraints on knowledge assets. There is a perceived operational time frame associated with each operation and any semantic integration solution that aims to have an impact must respect this. In today's semantic integration research, time constraints and availability of knowledge assets are not a high priority. To bridge this gap, we set out to explore the use of advanced semantic integration solutions, like ontology mapping [4], which have gained a lot of attention and momentum in recent years[3].

One of the issues with ontology mapping 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 [5], and storing and sharing alignments [6], the use of alignments in applications remains sparse.

One of the reasons for this phenomenon is the relatively high computational cost for processing ontology alignments. The W3C's[5] omnipresent `owl:sameAs` statement is only a starting point for what turns out to be a high computational cost and time consuming reasoning task to be undertaken by a Description Logic (DL) reasoner. That is because the `owl:sameAs` statement merely indicates the logical equivalence of the two referenced terms, but does not provide any further information about their provenance and semantics. To take full advantage of this logical equivalence we need to consider more than just the reporting of this fact.

A. *The case for operational ontology fragments*

Our experiences from ontology mapping [7] show that only a small fraction of the referenced ontologies are used in the produced alignments. This observation opens the way for an ontology alignment informed fragmentation task that meets the strict operational constraints needed in a distributed 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 relevant portions of the original ontologies but they are smaller in size and complexity, thus easier to process in a operationally useful time frame.

As an alternative some proposals to overcome the high computational cost and high latency in performing an inference cycle involve fragmentation or modularization of the original ontologies. Others approaches consider enhancing the existing OWL[6] vocabulary with richer constructs that enable more provenance information and semantics to be exposed when logical equivalence is reported. These however are heavyweight approaches that pose certain assumptions on the domain and applications.

We advocate a practical solution to this problem which is ready to use with current technology: to 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 rule 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 without affecting the original ontologies. These fragments can then be accessed and re-used at a lower computational cost than that of accessing and re-using the original ontologies.

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

B. *The POAF process*

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 has 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 logical equivalence of two OWL constructs (ranging from individuals to classes and properties, depending on the type of OWL language used). When automated reasoners encounter `owl:sameAs` or similar statements they 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 an 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 it is likely that unnecessary crawling of the OWL ontologies will occur (depending on the type of reasoning task). Even if the task is to simply resolve the name of the aligned concept, visiting the originating ontologies will add unnecessary time 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

[3]http://www.ontologymatching.org/ for an up-to-date overview of the field
[4]e.g. the SKOS vocabulary: http://www.w3.org/TR/skos-reference/
[5]The World Wide Web Consortium - http://www.w3.org/
[6]http://www.w3.org/2004/OWL/

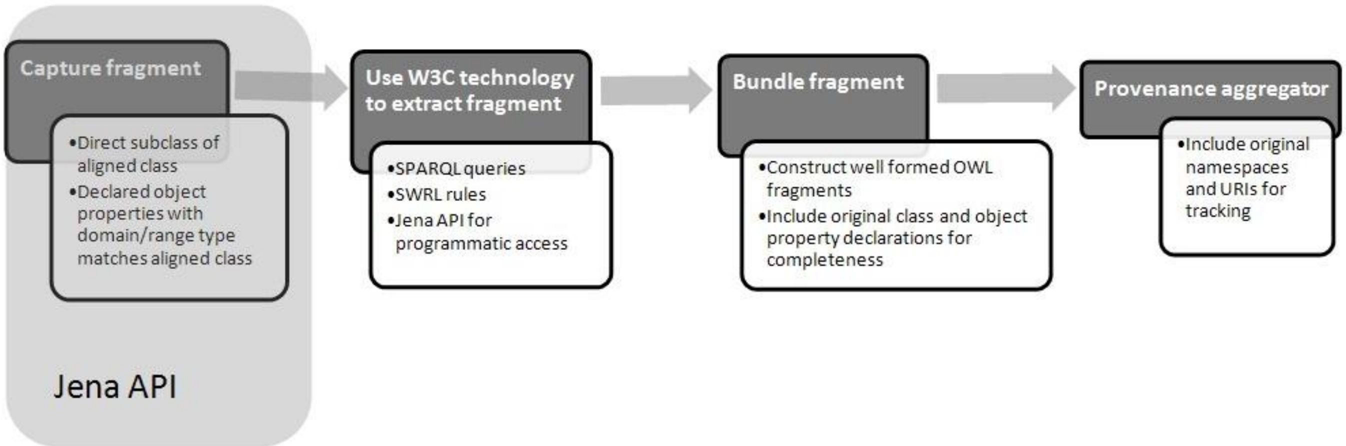

Figure 1: The POAF process

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 logical equivalence for the aligned concepts, it is difficult to track their provenance. This becomes increasingly problematic as more and more ontologies are involved. At the very least it is often not feasible to inspect the origin of the aligned concepts using eyeball checking in 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: those of unnecessary reasoning steps, overload of time and bandwidth resources, and knowledge base fragmentation.

Based on these observations, and our experiences with designing, developing, deploying and using ontology mapping systems [7], [8], 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 dub these fragments *POAF* (Portable Ontology Aligned Fragments) to highlight the tight coupling of the ontology alignment and the generated fragment. As the name indicates, we are interested in fragments of ontologies and, in particular, we aim to make those fragments portable. By that we mean:

1) Capture a fragment of each 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 that reasoners can use automatically as if they were mini-ontologies;
4) Include appropriate provenance information in these fragments, so that tracking and tracing of the original ontology is feasible.

These steps have been implemented in a process which operationalises the idea of *POAF* (See Figure 1).

The first step of the *POAF* generation 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[7] rules, and the Jena[8] Semantic Web framework from Hewlett Packard Labs. The next step is the *Bundle fragment* task. 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. In this task we collate the original namespaces with *POAF* specific ones to distinguish between declared and inferred statements in the *POAF* files. The *POAF* process is shown in further details in section IV through the form of a worked example which is used to capture and make available a specific alignment between two existing ontologies.

The following section provides an overview of the second technique covered by this paper: *SWEDER*.

III. SWEDER - Semantic Wrapping of Existing Data sources with Embedded Rules

A plethora of electronic data is available in structured forms today, either in existing Semantic Web encodings such as OWL/RDF or, more likely, in more traditional formats such as XML, CSV, HTML or relational databases. This data is available to the consumer, usually via URIs resolving to network or local addresses, but may also be sourced from directly referenced files and other non-URI referenced resources. The data itself can take any form, but we propose that it can be relatively easily semantically wrapped through a lightweight

[7]http://www.w3.org/Submission/SWRL/
[8]http://jena.sourceforge.net/

application of OWL/RDF to represent this existing data in the form of *entities* (classes), *attributes* (data properties) and *relationships* (object properties). The purpose of this lightweight semantic wrapping of these existing data sources is to provide representational homogeneity, thereby providing a common semantic basis for any downstream usage of this data.

Details regarding how this semantic wrapping is achieved are not discussed here since the focus of *SWEDER* is on how to leverage the semantically wrapped data sources. After the semantic wrapping has occurred *context ontologies* can be created to specifically capture the alignments between these semantically wrapped data sources, or potentially with existing ontologies from other sources. Representational homogeneity can now be assumed for all data sources that have been semantically wrapped in this way. In practical terms this semantic wrapping will likely be achieved through manual design and construction of a small ontology followed by a transformation process to convert the existing instance data to the new ontology. A similar existing pattern for this work is that of RDFa[9], microformats[10] (and GRDDL[11]) which are increasingly popular techniques for semantically enriching existing web-based data sources, albeit within existing web markup languages rather than as stand-alone ontologies as we are proposing.

A. *Context ontologies*

We adopt a dynamic notion of context which is not common to the formal notions presented in the AI literature (see, for example, the seminal work in [9] on formalizing contexts as first class objects). Our aim is to use context dynamically to capture each purpose for which data is used by a consumer application. It is assumed that the consumer application loads multiple data sources in order to fuse them or otherwise make use of these data sources and the relationships between them. We propose that the consumer application therefore adds a context to these data sources, since this specific combination of data sources has been selected to fulfil a particular need. *SWEDER* enables this context to be captured through the creation of a *context ontology* which specifically integrates the concepts from any semantically wrapped data sources or existing ontologies that it references. This *context ontology* can then be easily used by the consumer application, reading the various semantically wrapped data sources or existing ontologies, processing the instance data and executing embedded rules to derive further information or alignments. The result of this could be published as instance data conforming to the new *context ontology* and then made available for further usage by unknown downstream consumer applications.

An example of semantically wrapped data sources and their fusion using a *context ontology* is given in section IV.

Of course the fusion of two data sources by an application to achieve the results above could easily be achieved with existing technology, and does not require semantic representation. As a matter of fact, one could view the current trend of mashups[12] as a successful (usually non-semantic) form of such integrations. The specific benefits of semantically enabling the data sources and capturing the alignment representation and rules in a *context ontology* as defined in our approach lie in the representational homogeneity achieved through this technique, the self-defining and portable nature of the *context ontologies*, and the ease with which consumer applications can use them. Further important capabilities are also enabled through the use of this approach, most notably the support for referencing common definitions via URIs to enable more rapid understanding and information integration.

A key aspect of our proposal is facilitating the creation, representation and execution of information integration rules within these *context ontologies*, and this is something that existing OWL based solutions do not readily support. There are emerging standards in this area, notably SWRL (and potentially RIF[13]), but for various reasons we have chosen a lightweight, pragmatic approach and use SPARQL construct clauses to define and store these rules. This allows any SPARQL enabled endpoint to execute the rules and instantiate the inferred results directly from the construct clause without the need for any additional rule execution engine.

We store each SPARQL construct clause as instance data directly in the *context ontology* to which it applies, thus enabling these rules to be passed to the consumer application as part of the *context ontology*. In further iterations of this work we plan to introduce richer representation formats (such as SWRL, RIF) as these representations could generate SPARQL construct clauses which would be executed as per our current solution. The use of these richer representation languages would expose the semantics of these information integration rules to consumer applications rather than the current solution which simply records the text of the SPARQL construct clause without providing any actual representation of the rule composition.

B. *A conceptual model*

Conceptually and practically we use a two-tier model to represent each of our ontologies, both for the simple semantic wrappings of existing data sources, and for the subsequent *context ontologies* which are created to capture the alignments of other ontologies. This two-tier approach allows for a clear separation between the representation of the model and the capture of any associated rules. An example of this two-tier model is shown in figure 2.

The model ontology is at the centre, and it is comprised of entities, attributes and relationships, as described earlier. This is then imported into the outer rules ontology, which simply adds support for rules to be defined against the model. This two-tier approach is only needed when semantically wrapping existing data sources; existing ontologies can be used in their

[9]http://www.w3.org/2006/07/SWD/RDFa/
[10]http://microformats.org/
[11]http://www.w3.org/2004/01/rdxh/spec/
[12]http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)
[13]http://www.w3.org/2005/rules/

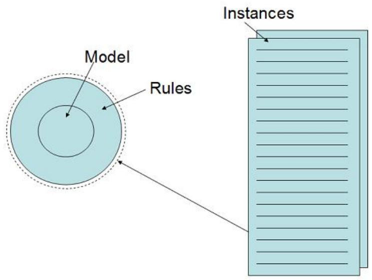

Figure 2: A two-tier ontology and associated instances

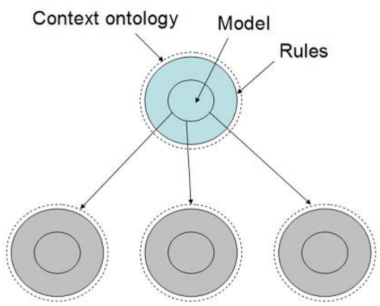

Figure 3: A typical context ontology

original form. In our current implementation the ability to define and store rules is provided through an import to a generic information integration rules ontology which enables the SPARQL construct based rules to be represented as simple entities, and records the subsequent instances of these rules as text in data-type properties. The separation of these two aspects of our ontologies enables the rules to be captured separately to the model, thus offering us a flexible way in which to improve the rule representation solution in the future without affecting the model ontology.

The final aspect of our solution is the capture of context information for multiple ontologies, which we achieve via the creation of additional lightweight *context ontologies*. These are built according to the same two-tier approach described previously.

The *context ontology* defines (in the inner model ontology) any additional entities, attributes or relations which are relevant to the current context, and also defines (in the outer rules ontology) any instances of rules which are used to populate these additional items. The *context ontology* also imports any required source ontologies (which may be existing ontologies, semantically wrapped data sources or *context ontologies*) and the new *context ontology* therefore captures the representation of this specific new context and embodies it in a semantic format consistent with the constituent ontologies. We visualize this approach in figure 3.

The final step is for a suitable consumer application to load the *context ontology* and any associated instance data for the source ontologies. Since all of the rules are contained within the *context ontology* this consumer application simply invokes

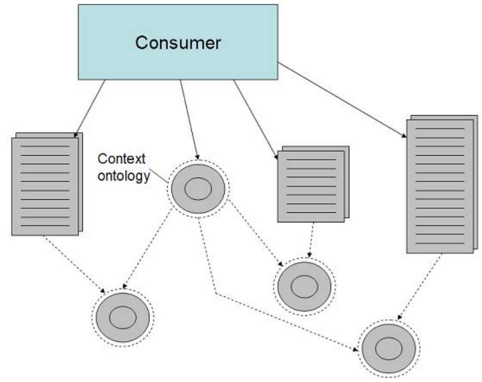

Figure 4: A typical consumer application

a standard process to extract all these rules (which are stored as SPARQL construct clause text), then executes these rules against an appropriate SPARQL endpoint. The results of these rule executions are that new instance data are created within the *context ontology*, and this can then be saved, published or further processed by the consumer application. We depict diagrammatically the interaction with a consumer application in figure 4. Note - the consumer application actually executes all the rules multiple times, until the set of all rule executions results in no further data being inferred.

The next section gives details of an example scenario in which both the *POAF* and *SWEDER* techniques are used together to affect a simple semantic integration solution between multiple data sources.

IV. An Example Scenario

A brief example scenario demonstrating semantic alignment and interoperability using the capabilities of *POAF* and *SWEDER* is outlined below. This example is brief, and in a real world scenario we would expect the complexity of the solution to be far greater, but still implemented using the basic *POAF* and *SWEDER* techniques.

We have a number of sources for this example, some are ontologies, and one is a structured dataset:

- `terrorism.owl`[14]: this is an OWL ontology that describes concepts in the terrorism domain and also contains instance data;
- `tkb.owl`[15]: this is an OWL ontology in the terrorism domain that describes the main concepts relating to terrorism attacks and also contains instance data;
- `World Gazetteer` list of towns and populations[16]: a structured text file containing a list of all towns and their populations (and other related information).

The high level summary of the proposed alignment and integration is based around cities and towns that are referenced in the ontologies and the datasets listed above.

[14] Available from http://counterterror.mindswap.org/2005/terrorism.owl

[15] An OWL ontology constructed for other ITA research work based on a published dataset at http://www.tkb.org, available on request

[16] Available from http://world-gazetteer.com

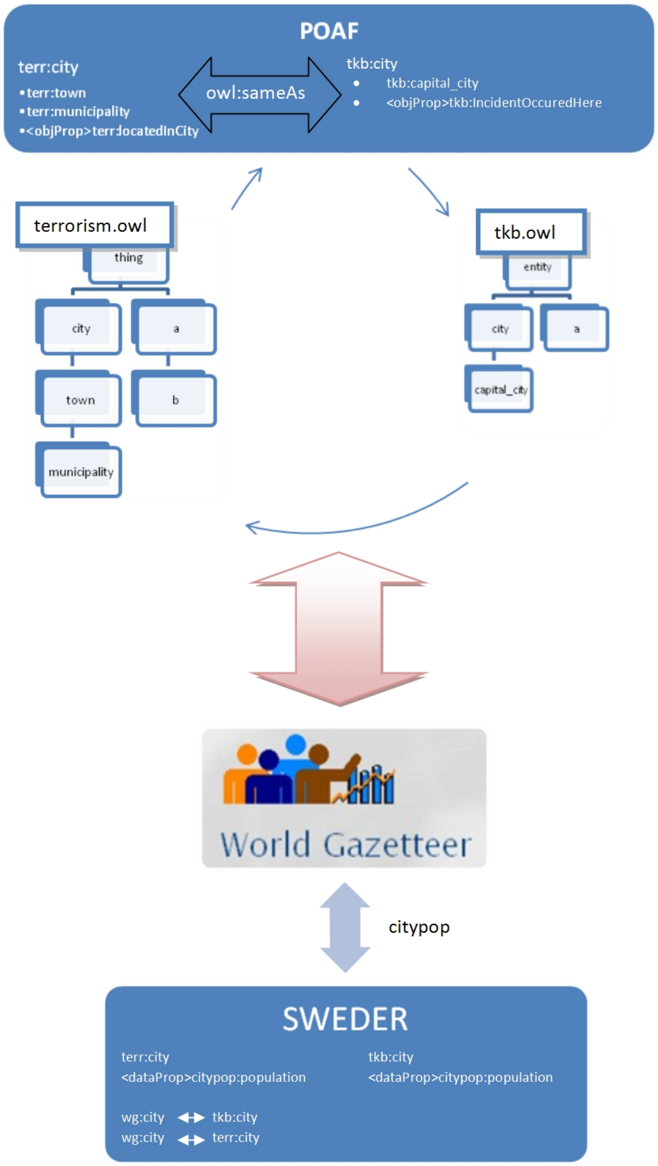

Figure 5: An example scenario

The first step in this example alignment is based on actual experiences in applying *POAF* in an experimental context in our ITA research activities. Full details of this example case can be found in the original *POAF* paper [1], but the relevant highlights are reproduced here:

An ontology mapping tool (CMS[17] - CROSI Mapping System) is used to determine potential alignments between the `tkb` and `terrorism` ontologies. Based on the specified matching criteria the CMS system identified an alignment between the `terr:City` concept in the `terrorism` ontology and the `tkb:City` concept in the `tkb` ontology. This logical equivalence is then able to be expressed using `owl:sameAs`.

The *POAF* process (See Figure 1) is used to encapsulate the appropriate fragments of the original ontologies to express this alignment along with appropriate provenance information:

- *Capture fragment*: The subclasses, data and object properties for the City concept in each ontology are automatically identified using taxonomic reasoning along with domain and range types respectively. This captures the direct provenance for each aligned concept and acts as an elaborate summary.
- *Extract using W3C technology*: SPARQL queries and SWRL rules are used to identify and capture the fragments identified in the previous step.
- *Bundle fragment*: This is achieved by either executing the SWRL rule(s) or running the SPARQL queries against an appropriate end point to capture the information in the *POAF* structure. This establishes relationships between the direct subclasses of `City` in each ontology in addition to certain relationships between object properties.
- *Provenance aggregator*: A designated `poaf:inferred` construct is used to store *POAF* inferred subclasses. In the future, all original namespaces and potentially other relevant provenance information will also be stored in similarly designated constructs.

The *POAF* process described above has encapsulated a specific alignment between these two ontologies, centered around the shared concept of `City`. The appropriate aspects of each ontology have been extracted (using SPARQL queries and SWRL rules) into a *POAF* file, along with some provenance information. The *POAF* file is now a stand-alone composite fragment of these two ontologies specifying the details of the alignment(s) and can be used either in future manual alignment review processes, or automatically by reasoners.

In this specific scenario this *POAF* fragment is then manually reviewed (along with many others) by a human analyst interested in taking a city-centric view of terrorist incidents from multiple data sources. This *POAF* fragment enables the analyst to identify that this alignment between the `tkb` and `terrorism` ontologies does indeed fuse information around the concept of `City`, and the provenance information included in the *POAF* fragment enables the analyst to validate the exact details of the alignment before accepting the alignment as correct and relevant.

The analyst wishes to list all known terrorist incidents by city, and this *POAF* fragment has provided full details of the mechanism needed to achieve this for these two ontologies. The analyst also has a requirement to list all known terrorist incidents by cities in particular population bands, e.g. cities with terrorist incidents and populations over 1 million, 5 million, etc. The analyst quickly identifies that (unsurprisingly) neither of the two existing terrorism ontologies contains details of the populations of the cities, although there is a very good resource of world city populations available from `World Gazetteer` in the form of a structured text file which could be used.

The *SWEDER* technique is used to create an ontological representation of the `World Gazetteer` data by creating a single class entity named `wg:City` with data and object properties for each of the attributes available in the original

[17] http://sourceforge.net/projects/ontologymapping/

structured text file (e.g. Name, AlternativeNames, Population, Type etc). This represents the *SWED* (Semantic Wrapping of Existing Data source) part of the *SWEDER* technique and results in a new `WorldGazetteer` ontology being made available along with instance data for every city contained in the original structured text file.

The next part of the *SWEDER* technique (*ER*: with Embedded Rules) is used to implement the alignment between all three of these ontologies through the use of rules expressed through SPARQL construct clauses, and certain property extensions to the various `City` concepts in each of the ontologies to contain the aligned results. This step is achieved through the creation of a new *context ontology* to capture these rules and these new properties. This *context ontology* is also used as the namespace within which any inferred instance data will be stored.

The specific rules and property extensions that are defined in this new `citypop` *context ontology* are:

- New property `population` defined on `tkb:City`
- New property `population` defined on `terr:City`
- SPARQL rule to identify matches (based on `Name` or `AlternativeNames`) between `wg:City` and `tkb:City` instances.
- SPARQL rule (as above) to identify matches between `wg:City` and `terr:City` instances.

These rules and property extensions are encapsulated within the new *context ontology*, and in conjunction with the semantically wrapped World Gazetteer data expressed in the `WorldGazetteer` ontology enable the population data to be directly added to the corresponding `City` instances for the `tkb` and `terrorist` ontologies.

Note - it is assumed that a DL reasoner will be used to infer the logical equivalence of the `tkb:City` and `terr:City` instances, however if there is no desire to use a DL reasoner for this then further SPARQL rules and properties can be constructed in the new *context ontology* to capture relationships between the `tkb:City` and `terr:City` instances. The manual construction of these SPARQL rules and the required properties is informed by the contents of the *POAF* fragment previously described.

The final step in the process is for a consumer application to load the three relevant ontologies (`tkb`, `terrorism` and `citypop`) and the appropriate instance data. The `citypop` ontology contains the SPARQL construct rules which enable the corresponding population figures for each city to be extracted from the `WorldGazetteer` ontology and represented in new `population` properties on both the `tkb:City` and `terr:City` corresponding instances. These rules are run simply by extracting the SPARQL construct texts from the `citypop` ontology and executing them against an appropriate SPARQL endpoint. Any resulting new instance data will be persisted in the namespace of the `citypop` ontology, but as properties against the `City` entities of the `tkb` and `terrorism` ontologies. This means that any future usage of the original `tkb` or `terrorismn` ontologies will not yield this additional population data, but loading of the `citypop` ontology will load `tkb` and `terrorism` ontologies with the population properties and data previously inferred. This enables the consumer application to fulfill the analysts requirement to see terrorist incident data against cities of certain population bands. An illustration of this combined *POAF* and *SWEDER* integration and alignment process is depicted in Figure 5

This example has highlighted the power of the *POAF* and *SWEDER* techniques in support of semantic integration and ontology alignment. Whilst the example is simplistic it does demonstrate two separate techniques to achieve alignment between three ontologies from both a model and instance data perspective, including the persistence of the rules and any instance data inferred as a result.

The next section gives details of other work which is related to the *POAF* and *SWEDER* techniques.

V. Related work

Euzenat and colleagues use the notion of ontology alignments to inform the syntax for expressing ontology modules [10]. Their main aim is to use ontology alignments in order to take advantage of the alignment composition features embedded in them, allowing 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 the use of ontology alignment within *POAF* which is to identify the starting points for the fragmentation algorithm which employs standard taxonomic reasoning.

Taxonomic reasoning is similarly deployed to *POAF* in [11]. The authors propose a modularization algorithm and a set of requirements for ontology modularization. The algorithm takes advantages of the subsumption checking of Description Logics (DLs) to identify self-contained pieces in the given ontology.

Others argue that relaxing the subsumption dependencies between ontology constructs can improve the expressivity of modular languages [12]. They argue for new modular semantics or extensions to improve the functionality of `owl:import` whereas POAF uses only existing semantic representation constructs.

Different notions of contexts have been proposed and investigated in the past. For example in [13] the authors argue for different types of contexts that contribute information relevant to natural language understanding. Each context is used to serve a different purpose, similar to our *SWEDER* work where we adopt a dynamic notion of context that is closely related and dependent on the use of source data. Our work uses source data and a set of semantic wrappers to elicit context and represent it in lightweight ontologies. Similarly, context has been used in [14] to aid in ontology elicitation whereby certain features of context dictate the primitive ontological constructs that will comprise an ontology.

Information integration, the driver behind our *SWEDER* work with semantically wrapped data and *context ontologies*, is also the focus of [15] but the authors deploy different means to achieve that: they propose to use a special kind of context knowledge, namely assumption knowledge, which refers to a set of implicit rules about assumptions and biases that govern the source data. This is similar to our notion of rules that integrate information from semantically wrapped data but we apply them at a later stage.

Another interesting angle we investigate with the use of *context ontologies* is deploying rules to capture the dependencies between properties. This is similar to the work of [16] where the authors elaborate on a naming convention scheme which is based on a loose ontology that represents the notions of kind and super-kind. Their aim is to ease data usability by providing a naming scheme that allows for classification of source data. In our *SWEDER* work we use properties and super properties found in the *context ontologies* to aid information integration and grouping.

VI. Conclusion

In this paper we have described two techniques to aid interoperability and semantic integration in a distributed network environment: *POAF* and *SWEDER*.

POAF (Portable Ontology Aligned Fragments) is a lightweight mechanism for sharing and distributing enhanced ontology alignment information. Making use of the generated alignments between two ontologies and exposing their immediately related terms in a well defined and concise OWL structure. *POAF* taps into the area of ontology fragmentation, but is not directly related to a fragmentation strategy. Rather, it highlights the importance of distributing more information when reporting logical equivalence between two aligned terms. This information could enhance uptake and re-use of ontology alignments. *POAF* may also contribute to adoption of Semantic Web technologies in scenarios where operational time constraints and resource load are important considerations. A *POAF* based solution aims to reduce the time needed to process ontologically aligned structures as it encapsulates the aligned concepts and their immediately related concepts in small, portable, 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 validate the proposed alignment. This process could be manual or semi-automatic depending on the assessment task.

SWEDER (Semantic Wrapping of Existing Data Sources with Embedded Rules) is a pragmatic approach to semantically enable existing sources of data, and then utilise multiple semantically enabled sources of such data through the creation of *context ontologies* to capture the specific rules and any new entities, relationships or attributes arising from this new context. *SWEDER* allows the storage of rules directly within the ontologies in such a way that they can be easily extracted and executed by a common capability within any consumer application, specifically through the usage of SPARQL construct clauses.

The example in section IV gave a brief worked example of how *POAF* and *SWEDER* can be used together to address different aspects of semantic alignment and integration.

Acknowledgment

This research was sponsored by the U.S. Army Research Laboratory and the U.K. 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 author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] Y. Kalfoglou, P. Smart, D. Braines, N. Shadbolt, *POAF: Portable Ontology Aligned Fragments*. In International Workshop on Ontologies: Reasoning and Modularity (WORM-2008) hosted by the 5th European Semantic Web Conference (ESWC-2008). Tenerife, Spain (2008).
[2] D. Braines, Y. Kalfoglou, P. Smart, N. Shadbolt, J. Bao, *A data-intensive lightweight semantic wrapper approach to aid information integration*. In 4th International Workshop on Contexts and Ontologies (C&O) hosted by the 18th European Conference on Artificial Intelligence (ECAI'08), Patras, Greece (2008).
[3] 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.
[4] Y. Kalfoglou and M. Schorlemmer. *Ontology mapping: the state of the art*. The Knowledge Engineering Review, 18(1):1-31, 2003.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] J. McCarthy, *Notes on formalizing contexts*. In Proceedings of the 5th National Conference on Artificial Intelligence, Los Altos, CA, USA, pp. 555560, (1986).
[10] 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.
[11] 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.
[12] 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.
[13] R. Porzel, H-P. Zorn, B. Loos, and R. Malaka, *Towards a separation of pragmatic knowledge and contextual information*. In Proceedings of the 2nd International Workshop on Contexts and Ontologies (C&O 2006), Riva del Garda, Italy, (August 2006).
[14] P. DeLeenheer and A. deMoor, *Context-driven disambiguation in ontology elicitation*. In Proceedings of the AAAI05 Workshop on Contexts and Ontologies (AAAI05/W1), USA, (July 2005).
[15] H. Zeng and R. Fikes, *Extracting assumptions from incomplete data*. In Proceedings of the CONTEXT05 Context Representation and Reasoning (CRR05), Paris, France, (July 2005).
[16] N. Cohen, P. Castro, and A. Misra, *Descriptive naming of context data providers*. In Proceedings of the CONTEXT05 Context Representation and Reasoning (CRR05), Paris, France, (July 2005).