Finding common ground: integrating the eagle-i and VIVO ontologies

Carlo Torniai[1,*], Shahim Essaid[1], Brian Lowe[2], Jon Corson-Rikert[2], and Melissa Haendel[1]
[1] OHSU Library and Department of Medical Informatics, Oregon Health & Science University, Portland, OR, USA
[2] Library, Cornell University, Ithaca, NY, USA

ABSTRACT

This paper describes the approach and strategies we have been applying during the process of integrating and modularizing the eagle-i and VIVO ontologies into the Integrated Semantic Framework. As this effort is yet on going, we will subsequently provide an evaluation report when the merger and modularization are complete. We welcome insight and comments from the ontology community about our progress and approach thus far.

1 INTRODUCTION

The eagle-i (www.eagle-i.net) and VIVO (www.vivoweb.org) projects are two large consortial efforts funded simultaneously by NIH from 2009 through 2012 with the goal of representing and cataloguing research resources and researcher profiles respectively. Both projects had developed ontologies to drive their respective search portals, data ingest processes, and tools. The eagle-i and VIVO ontology teams began coordinating efforts in 2009 to address overlapping areas of interest. We recognized that there was an artificial split introduced by the funding process between managing research resources in eagle-i and researcher profiles in VIVO. Such a division would impede the desired long-term goals of improved research networking within a single institution or across many. The VIVO and eagle-i ontology teams received NCATs funding in 2012 to merge our ontological efforts into one ontology suite in the context of the CTSAconnect project (www.ctsaconnect.org). The goals of CTSAconnect are first, to merge the eagle-i and VIVO ontologies into one single ontology suite (the Integrated Semantic Framework, ISF); second, to extend coverage to include representation of clinical encounters; third, to refactor the ISF such that it can be made available in a set of modules that can be reused independently; and fourth, to develop a data model and algorithms for computing practitioner expertise, and publishing it as Linked Data. The process of integrating the eagle-i and VIVO ontologies, refactoring them, and modularizing the ISF posed a set of interesting challenges and constraints:

1.1 Ontology principles vs. use cases

While the two ontologies have a lot of overlapping content, they emerged from very different modeling approaches. eagle-i uses the Basic Formal Ontology (BFO) (Grenon *et al*. 2004) as upper ontology and follows the OBO Foundry principles (Smith *et al*.,2007) for ontology engineering (numeric URIs, reuse of existing OBO Foundry ontologies through the MIREOT (Courtot *et al*., 2009) principle, etc.). Moreover, a unique feature of eagle-i was the effort to develop an ontology capable of driving an application directly while remaining for the community a valuable and reusable ontology covering research resources. This goal was accomplished by having a clear separation between application-specific vs. domain content (Torniai *et al*., 2011).

Starting from an existing production system, the VIVO team sought to generalize its internal ontology by selectively adopting ontologies already in wide use across the Linked Data community such as FOAF (xmlns.com/foaf/spec/) and BIBO (bibontology.com), without initially aligning to any upper ontology framework. The team sought to represent the structure of academic and research relationships (including related organizational entities and activities) within a VIVO core ontology while relying on references to external vocabularies to associate people, processes, or outcomes with the most appropriate domain of research. The usage of OWL axioms in VIVO primarily supported consistent data entry through generated UI templates rather than serving a role in classifying domain content. eagle-i also used the OWL language to drive the application and UI, but differently from VIVO this use was kept separate from the OWL axioms that define the semantics of the domain content.

1.2 Active application development and adoption

The eagle-i and VIVO ontologies and their respective applications are still in active development. To some degree this provides both teams the flexibility to improve their applications in response to improved ontology designs, but the applications must also respond to other requirements.

Both VIVO and eagle-i have been implemented at multiple production sites with significant amount of generated data. The need to support production systems with a minimum of downtime requires a very flexible refactoring approach that can easily support changes to the source ontologies and related applications. The need to provide clear update and data migration paths has also posed constraints on the development of the ISF in terms of maintaining data integrity over time without breaking key application functionalities. However, existing VIVO and eagle-i data will be used as benchmark to evaluate the effectiveness of ISF.

* To whom correspondence should be addressed: torniai@ohsu.edu

2 APPROACH

By aligning and integrating eagle-i and VIVO ontologies, the CTSAconnect project aims to produce the ISF as a set of distinct modules (see ontology source at http://code.google.com/p/connect-isf) that will replace eagle-i and VIVO ontologies in these applications and can also be used independently or in combinations by other applications across multiple domains of research. These applications will address, in different and sometimes unique ways, the management and discovery of research resources, research outcomes, research projects, expertise, and the full range of attendant human, scientific, and organizational resources. Even within our respective consortia, eagle-i and VIVO installations have fulfilled different needs in terms of content coverage. Some institutions have implemented VIVO as researcher profile systems but also include descriptions of eagle-i related resources such as core laboratories or equipment, while some eagle-i installations have felt the need to include researcher affiliations and organizational hierarchies to represent the complexity of their structure. Providing a suite of modular ontologies rather than a monolithic owl file will allow developers to integrate only required ontology modules in their applications and will also facilitate adoption of ISF modules as a reference point for later domain-specific ontology extensions.

2.1 Incremental approach

We started our integration process with two initial objectives: first, to identify the overlapping and duplicated entities in the eagle-i and VIVO ontologies, and second, to avoid severe disruptions in application compatibility. Though both main application teams are willing to invest in application improvements as necessary, the ontologies are also in use by other applications whose developers may not be prepared to make similar commitments. Therefore, we initially chose a conservative approach focusing on performing minor incremental additions to the ISF and pushing any other significant changes back to the source ontologies repositories for implementation. This initial approach worked well for referencing existing entities while developing new ISF-specific modules, and for performing initial alignments on classes in some portion of the overlapping hierarchies (such 'organization' and 'service' hierarchies). The lengthy process of identifying necessary alignments, implementing the changes in the source ontologies, and checking that no disruption was caused to the applications, became too limiting when exploring new design patterns or axiom-level integration.

This conservative, non-disruptive approach could have been more successful if the two source ontologies were more orthogonal and had used common reference ontologies, or if the proposed new content was orthogonal to existing content and we had no constraints imposed by applications already using the original eagle-i and VIVO ontologies. However, we later found a number of areas where departing from both existing ontologies allowed a more granular refactoring of existing entities and axioms, and freed the combined team to explore novel design patterns that would prove useful to both applications. When looking from both application perspectives, similar structural patterns could be recognized more clearly, and often times we were able to create something better than either ontology had previously developed. In these cases, we felt it would be more efficient to implement these changes directly in the ISF rather than incrementally disrupting the source ontologies and their applications.

2.2 Refactoring approach

The decision to abandon a conservative, incremental approach and depart from syncing changes with existing eagle-i and VIVO ontology repositories required the creation of new OWL files that hold the new, reused, and refactored content. These files collectively form the base for the refactored and modularized ISF. This new approach also required the ability to make entity and axiom level choices and mappings, being able to review these choices with the full CTSAconnect development team, and finally to derive the new stable OWL files during this process.

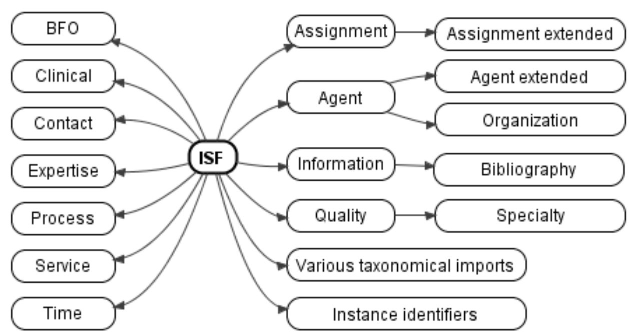

Fig. 1 The main ISF modules. They reflect the scope and nature of the content from eagle-i and VIVO that is being merged into the ISF.

The ISF ontology is currently taking shape as a set of small and relatively independent core modules with additional extensions where needed (see Fig. 1). Each core module is an OWL file that contains the relevant entities, axioms, and annotations, while other dependent modules reference needed entities and axioms without having to import the full module. This approach clearly identifies modules of reusable content to facilitate adoption or reuse by other applications and any future projects, while keeping eagle-i and VIVO specific content separate for optional consumption depending on the nature of the application.

Several modules blending eagle-i and VIVO requirements have already been developed, covering the broad areas of training and credentialing (included in the assignment module), contact information (contact module) and organizational positions and roles (included in the agent and assignment modules and their extensions). This effort also led to a parallel longer-term effort with other ontology developers to formalize the modeling of entities that have a clear social and organizational aspect and that do not have a clear place under the current BFO hierarchy.

Increasing the complexity of the ontology merging process created more impetus to keep track of changes and document and validate them. To this end, we developed a Protégé plugin that better supports this new process (see Fig. 2). The plugin provides a small Protégé view to add "refactoring" OWL annotations with structured fields that capture decisions and notes regarding the reuse of an entity or axiom, the mapping of an entity to a preferred one, and the assignment of alternative labels. It also provided visual feedback in Protégé to indicate which entities and axioms have been reviewed, selected, deprecated, etc. The annotations also specify the module name for the refactored OWL axioms. These annotations are then used in two main scripts. The first script generates the OWL files based on the decisions reflected in the refactoring annotations and the second script creates a spreadsheet summary of the refactoring changes to better support the review and approval process.

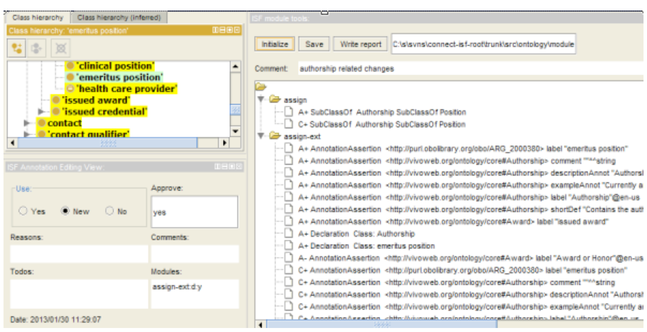

Fig. 2 The Protégé plugin. The left side shows the annotation view with entity highlighting and the form for an annotation. A green highlight shows an approved entity change and yellow indicates pending approval. The annotation form simplifies the editing of the structured refactoring annotations. The module view on the right side shows pending axiom changes per module with the ability to save the changes with a log comment, and to generate the spreadsheet summary report.

As an example, one of the first modules that was created according to this new workflow was a "contact" module that models contact-related data for any system that adopts the ISF. Both eagle-i and VIVO had very different types of OWL entities to capture contact data and a significant portion of the data was in the form of data properties and unstructured text. We examined two existing options for our new contact model and we decided to reuse and extend an existing RDF implementation of the community standard VCard specification. The VCard specification represents contact-related data in a very granular (when needed) fashion with opportunities for extensions and has been widely adopted in other domains, bringing also the potential to streamline data ingest and publishing procedures. In this model, most contact related data elements are represented as OWL classes and object properties instead of simple data properties. This mode of representation could help with later integration with other information ontologies such as the Information Artifact Ontology. The grouping of several optional properties found together in one context as a VCard object will also provide a means of maintaining alternative name and affiliation variants for a person or organization over time. This is a requirement that has emerged from practical experience in maintaining large applications and from anticipation of the need to track researchers and their activities as they migrated from one institution to another over the course of their careers. This granular model allows an entity to have a stable contact identifier while the components of the contact (physical addresses, phone numbers, web profiles, etc.) change or accumulate over time. Fig. 3 shows how this alignment was achieved in the context of the contact module.

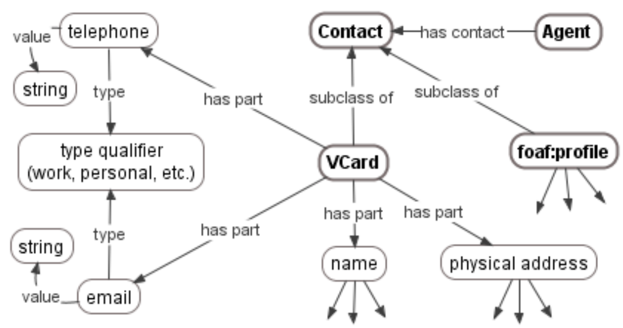

Fig. 3 "Contact" entity with VCard and foaf:profile specializations. eagle-i and VIVO agent contact properties (not shown) were migrated to this new model. The "contact" is then related to a person or organization (Agent in figure).

3 DISCUSSION AND FUTURE STEPS

The ISF development and integration process has somewhat increased the size of the resulting ontology while greatly streamlining its use of properties. The initial eagle-i and VIVO ontologies contained 25452/217 classes and 169/313 properties respectively. As of April 2013, the refactored ISF contains 26464 classes and 306 properties across 16 modules. The high number of classes is due to the use of external taxonomies such as MeSH and ICD codes.

The process of developing the ISF has made it clear that a successful ontology integration project needs to conduct a detailed and comprehensive examination of the modeling decisions employed in each of the original ontologies before designing a merge plan. The complexity of the subsequent merge depends on the following factors: the level of overlap (or lack of orthogonality) in the concepts defined in the initial ontologies, the complexity added if the ontologies do not share a common upper ontology, and the degree to which areas of extensive overlap in content have been modeled with different approaches.

Where the anticipated complexity is high, due to these factors or to needs expressed via new use cases, the ontology merging process will likely require significant periods of ontological analysis, design pattern prototyping, and searching for existing solutions in other ontologies or data standards. This is a significant risk factor that has to be examined early in the project and some limits have to be set to avoid project delays. We have found that a careful modular approach also allows some of these issues to be addressed independently of other work, enabling progress on one front while more free-ranging discussions distill into consensus in challenging areas such as defining roles for organizations and people.

The CTSAconnect team has identified a number of avenues for future exploration, including collaboration with a group at the University of Arkansas for Medical Sciences that is working on socio-medical ontology issues. As this effort attempts to harmonize solutions produced by several community efforts (starting with those in the OBO Foundry), the large base of adopters of the eagle-i and VIVO applications will likely express needs that differ significantly from those of the biomedical research community.

3.1 Tool enhancement

We will work on enhancing our Protégé plugin in order to facilitate the integration progress. In particular, we want to provide functionality that allows direct connection between class or properties and data (triples) that use that particular class or properties in any statement or annotation. If this information is summarized while hovering over a class within Protégé it will give an immediate feedback of the impact of ontology changes on the data. We are thinking of implementing something similar to the "usage" tab but related to a particular data source.

3.2 Rules to support data migration

The data migration implications of ontology changes can be mitigated by the use of rules capable of adding or removing data elements to conform to new ontology patterns. The OWLIM triple store (http://www.ontotext.com/owlim) includes a rule engine that has shown promise in initial experiments for performing automated data migration. While the OWLIM rule language is not expressive enough to describe a complete mapping between the VIVO ontology and the emerging ISF, it is especially attractive for its ability to infer new anonymous resources. The creation of new resources is frequently necessary in migrating from existing RDF graphs to those compliant with the ISF, and is not supported in "DL-safe" rules languages such as SWRL. Further experimentation is necessarily to determine whether anonymous resources will prove adequate in the ISF based RDF consumed as linked data; they will likely be less troublesome in direct queries to SPARQL endpoints. While a rules-based approach does not obviate the need to modify the eagle-i and VIVO applications in the longer term, it may facilitate a period of transition where both applications can provide ISF-compatible data for crawling and querying while changes to the applications are in progress.

3.3 Generalization of approach

While most of the strategies and tools were developed for our particular needs, we think that they could be adapted and reused in other situations that call for comparing and integrating different ontology sources. The OBO Foundry, for example, has as a goal to create a suite of orthogonal interoperable reference ontologies in the biomedical domain. "Orthogonal" in this context means that there should be one identifier for a specific semantic entity and this identifier should be reused consistently across the OBO Foundry ontology suite. We think that the methodologies and tools we are developing could prove handy within the OBO Foundry for assessment and evaluation purposes, or for assembling new combined ontologies out of existing ontologies. The Protégé plugin could be used with two ontologies under evaluation to generate a set of annotations that would later trigger automatic changes and enable automatic report generation. We would like to collect through our tracker (http://code.google.com/p/connect-isf/issues/list) use cases and requirements that could help to refine and broad our approach and tools to meet other needs within the Biomedical Ontology community.

ACKNOWLEDGEMENTS

The work has been funded by NCATS through Booz Allen Hamilton. Grant Number GLIBY0020A.

REFERENCES

Grenon, P., Smith, B. (2004), SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition & Computation: An Interdisciplinary Journal* **4,** 69-104

Smith, B., et al. (2007), The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol,* **25** (11), 1251-5

Courtot, M., Gibson, F., and Lister, A. (2009), MIREOT: the Minimum Information to Reference an External Ontology Term. *ICBO*

Torniai C, Brush M, Vasilevsky N, et al. (2011), Developing an application ontology for biomedical resource annotation and retrieval: challenges and lessons learned. *ICBO*