Integrated Semantic Framework (ISF) ontology documentation for ISF release 1.0 dated 7/31/2013

Table of Contents:

Introduction

In this document, we present a description of the content, the design, and the released files for the Integrated Semantic Framework (ISF) ontology. This is the documentation of the first release of the ISF (1.0) dated 7/31/2013. It provides an overview of the content of the ISF and identifies potential future steps for development.

The ISF ontology is an OWL ontology that represents the result of a merge of the eagle-i [1] and VIVO [2] ontologies with additional content to represent clinical activities and expertise. The ontology content provided by the eagle-i ontology is mainly concerned with representing research resources and activities at the laboratory/basic science level. It provides representations for instruments, techniques, services, specimens, organisms, reagents, etc. The eagle-i ontology is used for the representation and sharing of such data from various institutions and is accomplished by integrating the ontology into the eagle-i application [3]. The content from the VIVO ontology complements the above content by representing researchers' activities such as publications, grants, research projects, academic and organizational positions, etc. The VIVO ontology is also integrated into the VIVO application [4] to support the capture and sharing of this information between various institutions. In addition to these two main areas, the ISF adds a simple representation of clinical activities to support the creation of a clinical profile for clinicians based on their clinical encounters, providing a model whereby clinical/subject expertise can be inferred. This representation relates basic science research (the eagle-i based content), academic professionals (the VIVO based content), and clinicians profiled through the outcome of their clinical encounters (such as billing codes for diagnoses or procedures). The main goal is to enable the creation of multidisciplinary research teams from the three areas described above as well as making this data available for third-party applications.

The first version of the ISF ontology was developed as part of the CTSAConnect [5] project, funded by the National Center for Advancing Translational Science (NCATS). This project also provided few additional components where the ontology is the core data model but other ISF-based components are built on top. For example, as part of the CTSAConnect project we integrated data from the CTSA ShareCenter [6] application, provided ISF ontology-based linked data for clinical expertise from a few participating institutions, prototyped the integration of cross-application queries, and visualizations for expertise, etc.

The following sections describe the ISF ontology as it is released for version 1.0 for end-users. Later, we describe the internal structure that would be of interest to contributors and developers.

Overview of the ISF ontology

The content of the ISF ontology is represented in OWL, however, there are three distinct approaches for modeling the domain of interest within the ISF ontology, as detailed below.

OWL classes

The ISF OWL classes represent types or sets of things in the domain of the ISF (research resources, academic activities, organization and person roles, and clinical expertise). These classes can be instantiated when needed to capture specific instances in the domain. For example, a specific person will be an instance of the ISF class "person" and a specific instrument will be an instance of the class "instrument", and so forth. Each instance will have its own unique identifier but all the instances of a specific class/type share the intended meaning of the class/type.

The intended meaning of an ISF class should be interpreted as follows. The ISF adopts the Basic Formal Ontology (BFO) [7] as an upper ontology and this imposes constraints on the meaning of the OWL class in the ISF ontology. Describing the BFO is out of scope for this documentation, but the main constraint is that the OWL classes in the ISF ontology are intended to represent real existing types of entities from the domain of the ISF according to domain experts and established domain knowledge. In addition to this general modeling constraint, the intended meaning of ISF ontology classes is further defined through OWL axioms and textual labels and definitions. The class labels are the least significant part of the meaning of ISF classes and could change as needed (but not the IRI identifiers) as long as the intended meaning remains the same.

OWL instances

In addition to the OWL classes described above, the ISF ontology also provides instances of the above classes where needed. For example, the ISF includes an "organization" class (that can be subclassed if needed) to be instantiated for various organization instances (not subtypes) and the ISF creates a set of such instances for organizations that are important for the ISF framework and ISF based data. This does not mean that other instances cannot be created, it only means that we had the need for creating and sharing a set of instances as part of the ISF ontology. For example, we include in the ISF "organization" instances for the various medical boards. These instances could be used to relate providers, their specialties or credentials, or any other relevant data to specific medical boards.

SKOS instances

The above approach describes standard modeling of a domain at the level of classes/types and the level of individual unique instances, which is useful for capturing new data as instances of the types. However, in some cases there is a need for a simpler, single-level approach to reduce computational complexity and ease usage.

For example, during a clinical encounter a provider records a diagnosis by referring to a code (such as an ICD9 code), or referring to other coding systems to code additional aspects of the encounter. There are at least two options for representing the references to these coding systems. First, we can say that there is a class named "reference to ICD9 code" and then create unique instances of this class whenever a provider references a code. However, in the scope of the ISF, we felt this to be an unnecessary redundancy and did not see the need to create unique instances each time an ICD9 code is referenced. We felt that it is more appropriate to have a set of codes or identifiers that can be used where appropriate without instantiating OWL classes. Because of this, we adopted a second approach where we do not create the class such as "reference to ICD9 code" and replace this with a "mentions" relationship from the IAO ontology. This relationship is then used to relate each unique clinical encounter instance to one or more entities from the ISF SKOS-based vocabularies to capture the relationship between an encounter and an ICD9 code, or other coding systems.

The Simple Knowledge Organization System (SKOS) [8] takes the following approach for modeling a domain. A domain in SKOS is modeled as instances of the SKOS class "Concept" and these instances represent concepts in the domain where there is no need to also represent instances of the concept classes themselves. In other words, in SKOS, there is no need or formal way to capture unique instances of a specific concept. In this modeling approach, we only care about the concepts themselves and we are not interested in the instances. For example, we care about the concept “person” but we do not care about representing each individual “person”. We found this approach to be more useful for representing existing coding systems within the ISF framework and we adopted it for the representation of ICD9 and MeSH, and their links to the UMLS.

The SKOS approach also allowed us to address other ISF use-cases where we felt that it is more appropriate to model the domain of the ISF as a vocabulary instead of classes and instances (see miscellaneous topics below for examples). How to best use external vocabularies in an ontological framework was one of the goals of the CTSAconnect project and after some exploration we decided upon the above SKOS-based approach.

ISF ontology project

The ISF ontology is currently hosted as a Google code project with a Subversion (SVN) repository. The project site is: http://code.google.com/p/connect-isf/

The ontology releases are under a top-level directory in SVN named “release”. There is also a “trunk” directory that represents the development area. Released ontologies will be resolvable online and can be viewed or downloaded directly as described in the next section without needing a local SVN checkout. However, if there is a need for a local checkout, the following are the SVN commands for a local checkout of a top level, release 1.0, or development:

svn checkout http://connect-isf.googlecode.com/svn
svn checkout http://connect-isf.googlecode.com/svn/release/2013-07-31
svn checkout http://connect-isf.googlecode.com/svn/trunk

The ISF tools assume a top-level checkout and need an environment variable that points to this checkout. Please see a separate “readme” in the Eclipse project for more details.

ISF 1.0 release

ISF ontology releases are in the form of OWL2 [9] files in the ISF SVN repository [10] and are organized in subdirectories by release dates. For example, the ISF 1.0 release files will be in a subdirectory named “2013-07-31”. The IRIs for the released ontologies will resolve online through a PURL service [11] so that the ontologies can be retrieved without needing a local SVN checkout. This can be seen in Figure 1.

Figure 1. The ISF SVN directory structure

isf.owl

The isf.owl in the above image is the ISF ontology with an ontology IRI of: http://purl.obolibrary.org/obo/arg/isf.owl and a release 1.0 version IRI of: http://purl.obolibrary.org/obo/arg/release/2013-07-31/isf.owl

This file represents the ISF ontology content other than the SKOS component described above. It is the result of a merge of the source files that are maintained under “trunk” in SVN. This merged format is a convenient representation of the ISF as a single OWL file. However, we also provide an alternative packaging of parts of the ISF content as described in the modules section.

isf-reasoned.owl

This file is based on the isf.owl file above. The above file is reasoned in an OWL reasoner and any additional subclass or equivalence axioms are added. The ontology IRI is: http://purl.obolibrary.org/obo/arg/isf-reasoned.owl and the release 1.0 version IRI is: http://purl.obolibrary.org/obo/arg/release/2013-07-31/isf-reasoned.owl

isf-skos.owl

This file contains the ISF SKOS content as described earlier. The ontology IRI is: http://purl.obolibrary.org/obo/arg/isf-skos.owl and the release 1.0 version IRI is: http://purl.obolibrary.org/obo/arg/release/2013-07-31/isf-skos.owl

Modules

In addition to the isf*.owl files which are meant to contain the content of the ISF, we made available self-contained module files derived from the content of the ISF ontology for specific use cases. The idea is that due to the wide coverage of the ISF, we want to make available subsets of the content for end users that only need portions of the ISF for their specific purposes.

The module files do not contain any new IRIs or axioms and any exceptions to this will be clearly documented. The ISF maintainers will support a small set of modules that are thought to be widely useful, or that are used by existing ISF-based linked data applications to simplify adding the ontology content to data from such applications. These modules are currently generated by release tools that are based on an OWL file that indicates what should be included in a module.

These modules are meant to be user friendly and we provide documentation for their use. The following subsections document the current set of modules and corresponding diagrams. Due to their size, a small version is included in this document and a link to a full image from the SVN repository is below each figure.

general-module.owl

This module shows the main ISF classes and properties used to represent data such as contact information, online profiles, location, time, and other types of entities that might be needed in other modules or used independently of them.

The ISF adopts a developing RDF mapping of the VCard standard [12] and the friend of friend (FOAF) vocabulary [13] to represent contact and profile-like information. These vocabularies provide many classes and properties for this purpose and they overlap in some areas. The ISF does not try to make judgments about which one is better and which one should be used if there is an overlap. The idea is that the data has to be encoded in the specific vocabulary if it is intended to be consumed by clients that know the vocabulary. The ISF generalizes the idea of a contact and creates one top-level "contact" class that VCard and foaf extend.

The module diagram in Figure 2 shows the names and abbreviated identifiers for the relevant classes and the OWL file includes those classes and properties along with related annotations and axioms.

The ontology IRI is: http://purl.obolibrary.org/obo/arg/general-module.owl and the release 1.0 version IRI is: http://purl.obolibrary.org/obo/arg/release/2013-07-31/general-module.owl

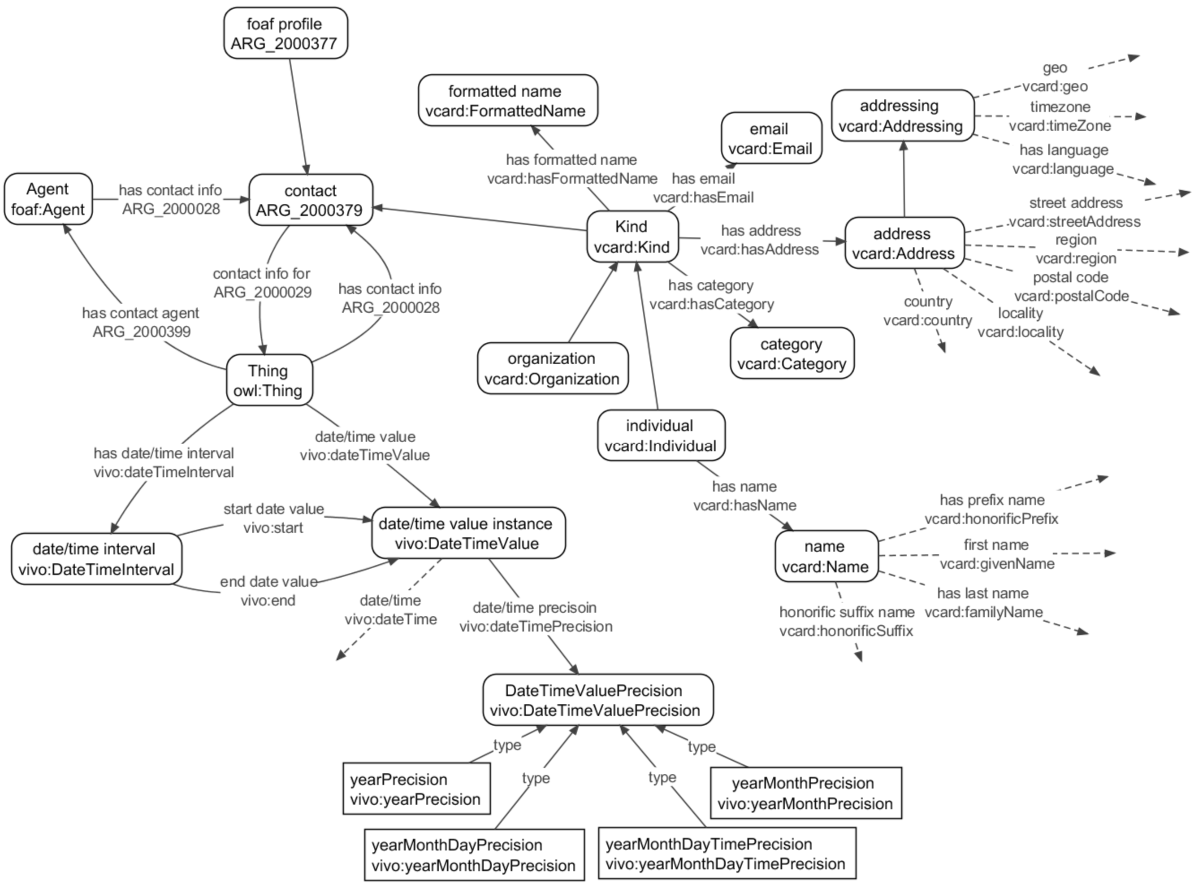

FIgure 2. ISF "general" module

http://connect-isf.googlecode.com/svn/release/2013-07-31/diagram-general-module.png

academic-module.owl

This module contains academic and researcher related entities, and is designed to be used in conjunction with the general module. It contains mostly content from the VIVO model that was merged into the ISF, which in turn includes classes and properties from the BIBO bibliographic ontology, FOAF, and the FAO geopolitical ontology.

Several of the most important classes in the academic module are children of the ISF "Relationship" class (See Figure 3). A common pattern involves the connection of different agents and their respective roles through a relationship instance linked to a date/time interval to indicate the time period in which the relationship holds. This pattern is used, for example, to relate persons to their departments or institutions through "Position" individuals, and to relate investigators, their research roles, and administering institutions through "Grant" and "Contract" instances. The roles related by these relationship instances may be realized in "Projects" or other processual entities. A similar relationship pattern is used with "Authorship" and "Editorship" instances to relate persons to publications while also specifying an author order ranking integer.

The ontology IRI is: http://purl.obolibrary.org/obo/arg/academic-module.owl and the release 1.0 version IRI is: http://purl.obolibrary.org/obo/arg/release/2013-07-31/academic-module.owl

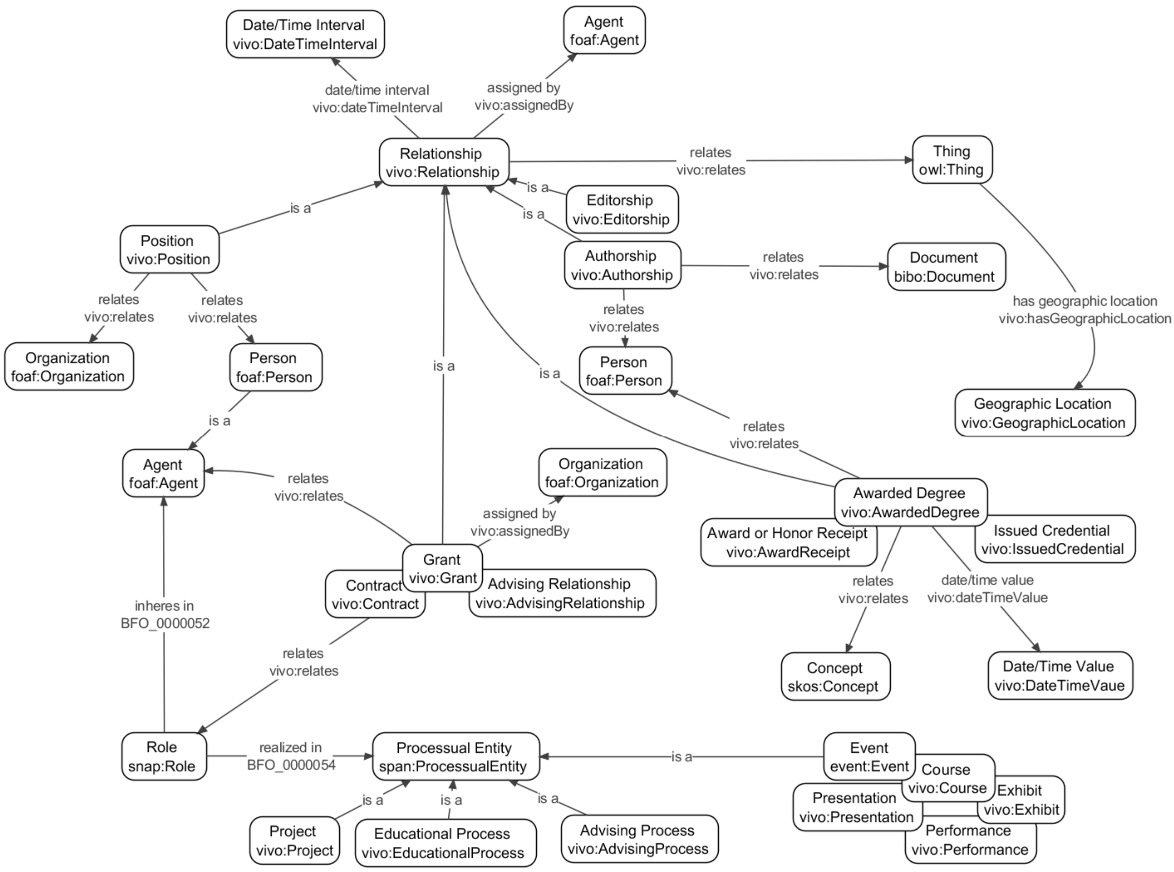

Figure 3. ISF “academic” module

http://connect-isf.googlecode.com/svn/release/2013-07-31/diagram-academic-module.png

research-module.owl

This module contains classes and properties necessary for representing research related resources (see Figure 4). By resources we mean the types of entities directly involved in research activities such as instruments, techniques, reagents, etc. Related “Person” and organizational related entities are largely in the academic module or can be added to a new module if needed. The content of this module is mostly derived from the merged eagle-i ontology.

The ontology IRI is: http://purl.obolibrary.org/obo/arg/research-module.owl and the release 1.0 version IRI is: http://purl.obolibrary.org/obo/arg/release/2013-07-31/research-module.owl

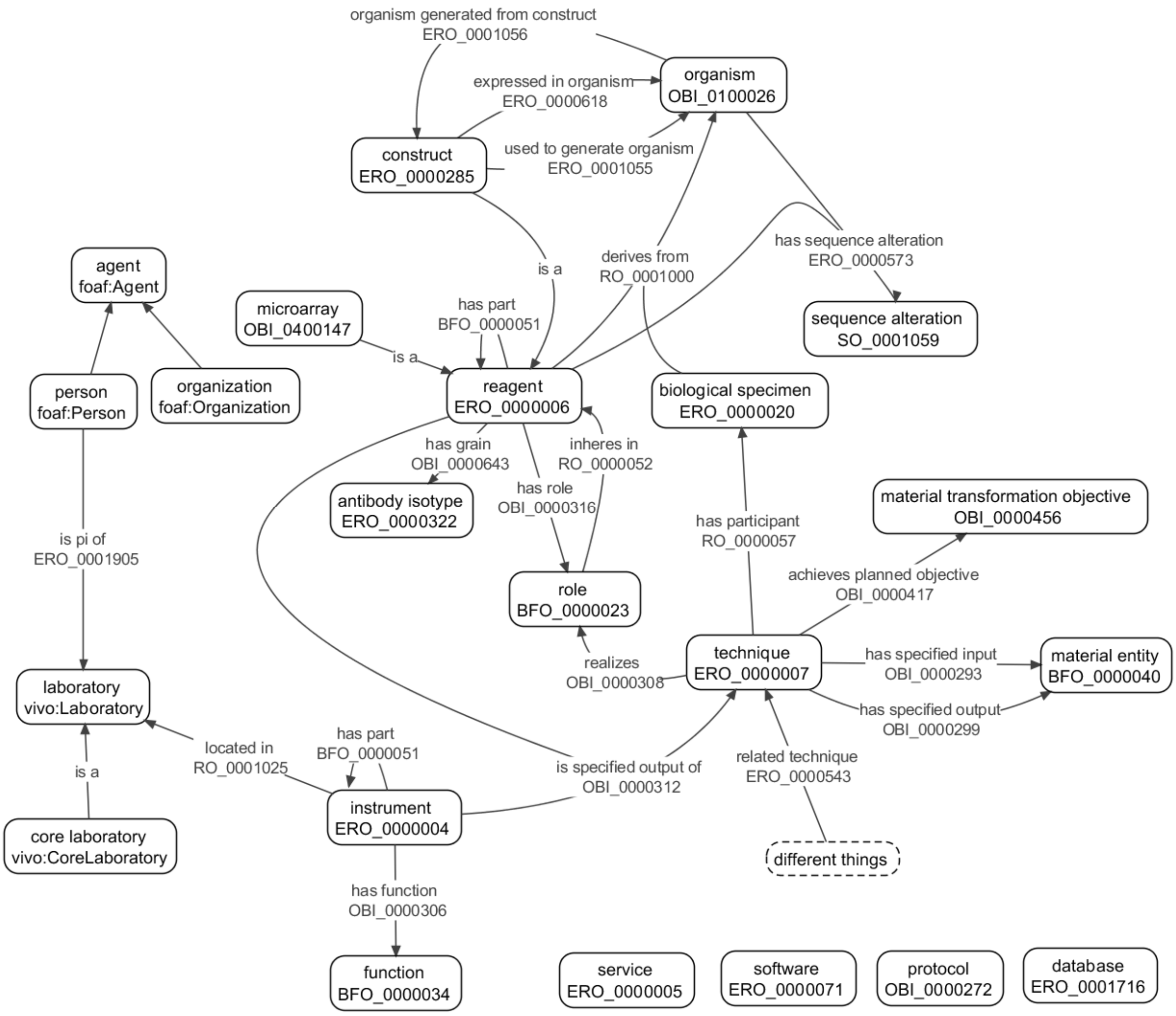

Figure 4. ISF "research" module

http://connect-isf.googlecode.com/svn/release/2013-07-31/diagram-research-module.png

clinical-module.owl

This is a relatively small module that contains a simplified representation of clinical providers, their clinical encounters with patients, and their specialties and credentials. It also contains classes and properties that can be used to measure the expertise of a provider. For example, we use this module to generate expertise measurements indexed by ICD9 codes based on clinical encounter data. The long-term goal is that we will be able to measure and represent various types of expertise measurements (based on different dataset and algorithms) with this module. We can then share and aggregate these measurements in order to support higher level applications that aim at connecting people and resources in a meaningful way. This module helps with profiling clinical providers in the ISF in order to link them to people and resources profiled with the other modules (see Figure 5).

The ontology IRI is: http://purl.obolibrary.org/obo/arg/clinical-module.owl and the release 1.0 version IRI is: http://purl.obolibrary.org/obo/arg/release/2013-07-31/clinical-module.owl

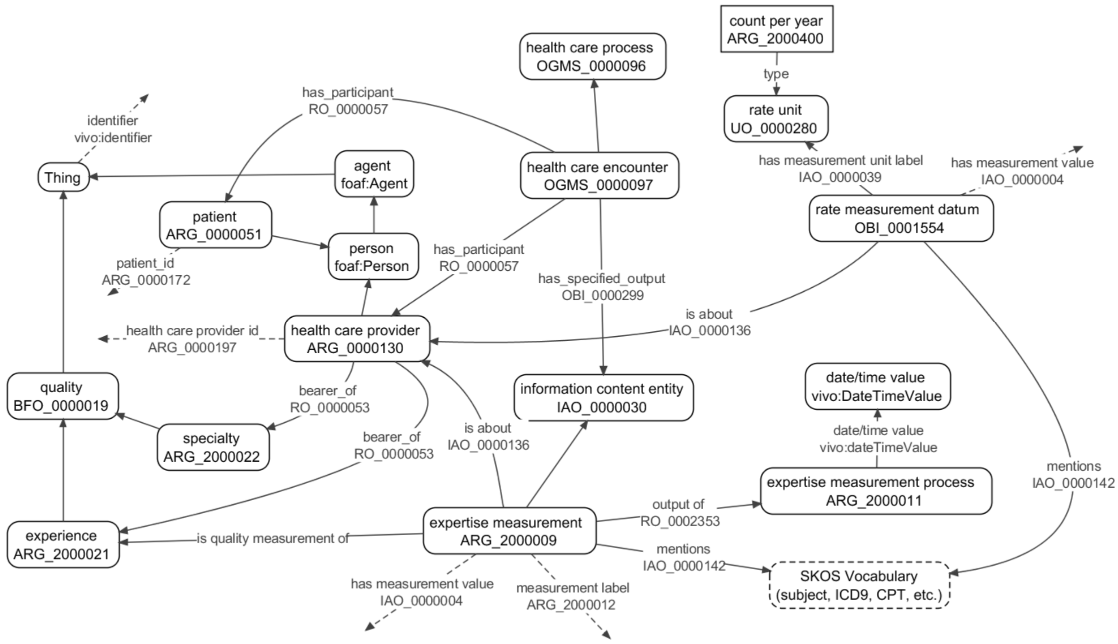

Figure 5. ISF "clinical" module

http://connect-isf.googlecode.com/svn/release/2013-07-31/diagram-clinical-module.png

ISF trunk

The trunk area is intended for ISF development and its structure might change to accommodate more efficient development workflows. However, Figure 6 shows the current SVN directory layout and it is expected to remain stable for the near future. Currently, we include catalog files so that the ontology IRIs for files in trunk point to the files available locally. To work with trunk, you need a local SVN checkout. However, there are few files that share the same IRI as the release files but the local SVN checkout, when used in Protégé, should resolve those IRIs to the local trunk files.

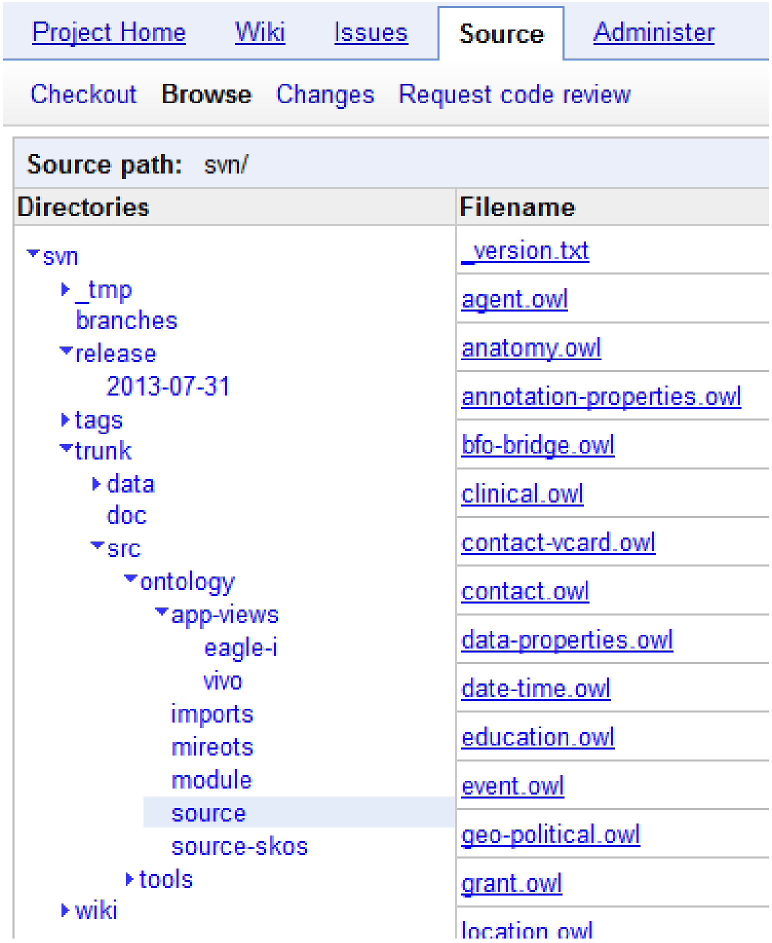

Figure 6. SVN “trunk” directory with files in the “source” subdirectory

The top directory for ontology content is the “ontology” directory. In this folder there should be an isf.owl file that is a top importer of other files that form the ISF ontology without the SKOS content. This file is the basis for the isf.owl release files and it provides a development version of the ISF. There should also be an isf-full.owl that also imports the ISF SKOS content from the isf-skos.owl file.

Adjacent to the “ontology” directory is a “tools” directory. This directory contains one subdirectory that represents an Eclipse Java project. This project contains various scripts that are used for developing and releasing the ISF. There is a “readme.txt” in the root of the Eclipse project that will describe how to import and use these tools. These tools are currently for internal use and will likely change. The are no library-like tools to use with the ISF at this time, but that will be a future direction.

The “doc” folder is an internal area for managing documentation content and the “data” folder will hold ISF related data such as demonstration data or other non-OWL/RDF data that cannot or should not be included in the ontology.

“ontology” subdirectories

The various subdirectories under “ontology” organize the ISF development area. Table 1 summarizes the current layout but this is an internal organization that could change as needed.

Directory	**Purpose**
app-views	Contains OWL files with additional ISF content that is only relevant to the corresponding application. There are currently two supported applications: the eagle-i and VIVO applications. This folder contains a set of application specific annotations (such preferred labels) and axioms that are related to User Interface presentation.
imports	Contains full copies of any ontology that is fully imported in the ISF.
mireot	ISF reuses some terms and properties from other ontologies that are referenced using the MIREOT [14] principle (see below). This folder contains the information about the MIREOTed entities organized by source ontologies. The previous MIREOT files from the ERO ontology were merged into the source files. New MIREOT files will be generated in a follow up minor ISF release based on the external entities that remained in the ISF.
source	The ISF ontology decomposed into (as much as possible) a set of non-overlapping files. The goal is that there should be only one file that defines an entity and also contains its annotations. Other files should only reference IRIs along with a copy their label. The file names in this directory indicate the focus of the file.
source-skos	The SKOS portion of the ISF vocabulary.

Table 1. Description of the folders under “ontology” in the ISF SVN

Miscellaneous topics

MIREOT: A MIREOT file is a file that contains some OWL content (classes, properties, annotations, etc.) coming from other existing ontologies (such Ontology for Biomedical Investigations (OBI) or the Gene Ontology (GO), that are used in the ISF ontology. This approach helps avoid having to import full ontologies and the scripted nature of this approach also helps with the maintenance of these files.

Ontology vs. application separation

Any application that uses the ISF ontology will likely need additional OWL content to support the specific application. This content can be extensions of ISF classes, preferred annotations, equivalence axioms, etc. The current SVN repository includes this content for the eagle-i and VIVO applications under the “app-views” directory. Other applications such as Plumage [15] would be welcome to include such application configurations in this directory.

One goal of including application-specific content in the ISF release is to be able to share, and reuse as much as possible, information between applications to avoid fragmentation in this domain. Also, some of this application-specific content can be included in future version of the ISF ontology if it is widely used and is considered to be ontological in nature.

External ontologies

In ISF we reused when possible entities form already existing ontologies. Most of the ontologies were referenced because of their terms were already present in the original eagle-i or VIVO ontologies. In addition to the above, the general criteria for choosing among the possible external ontologies we looked for a) Soundness of modeling b) Wide Adoption c) Defacto Standard for a specific domain.

ISF Vocabulary model

An ontology is mostly composed of classes of things (types, sets, etc.) from the domain of interest. These classes are then instantiated by creating class members (individual things) to represent the details of the domain. These individuals are given unique identifiers and are related to each other accordingly. However, there is also a need for being able to reference a concept in a way that is different from referencing individual instances. The following example clarifies this issue and describes our approach. This was one of the major goals discussed in the proposal - to develop a standardized mechanism to leverage non-ontological vocabularies.

In the ISF there is a class labeled “Awarded degree” and instances of this class are created when a degree is created or given to a person. The individual degree instances are uniquely identified for every degree given to a person. This class, in ISF, is meant to relate a unique degree to the organization issuing it, the person receiving it, time values, location, etc. (it is a representation of an n-ary relation), but instances of this class need to also specify the type of degree received. One option is to create additional classes in the ontology for every possible degree type (either as subtypes of “Awarded degree” or as a separate branch) and instantiate the type for each instance of an “Awarded degree”. This level of granularity was not found to be useful in our ISF modeling and our approach was to create a “Degree type” class and create a single shared instance for each type (vs. creating classes for each type). We consider these instances to be a vocabulary for the various degree types and this vocabulary can be used wherever there is a need to refer to a “Degree type”. An instance of “Awarded degree” is then related to one of the vocabulary entries to specify the type of awarded degree. The diagram in Figure 7 shows this approach.

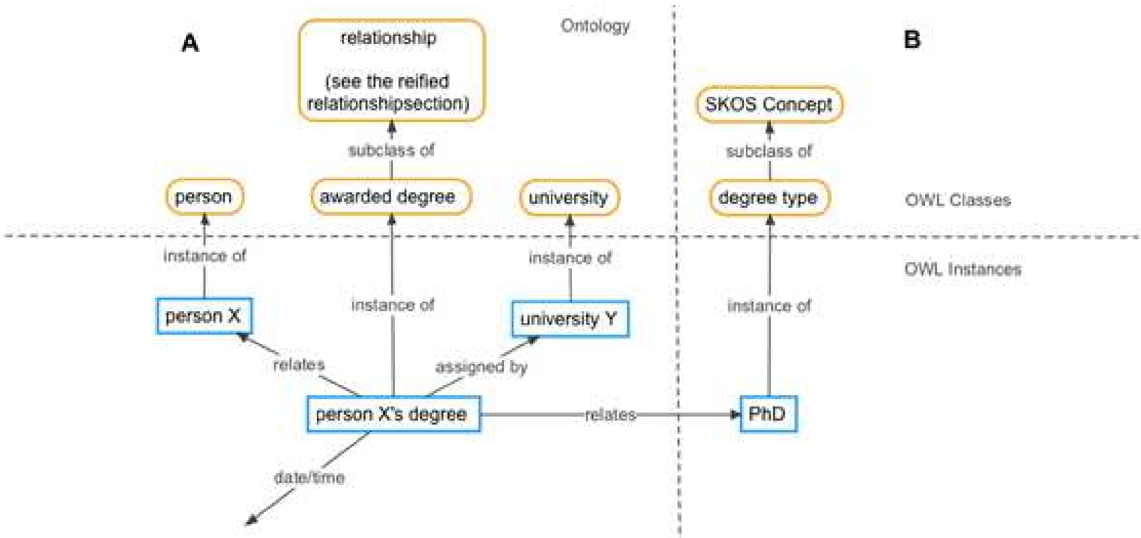

Figure 7. Use of vocabularies in the context of the ISF, illustrated using the awarded degree example. An instance of the "Awarded degree" class that belong to the "person X", which is an instance of the "Person" class. This awarded degree is also related to a specific university instance and to the vocabulary instance "PhD" that is an instance of the SKOS concept "degree type" (see the SKOS and "reified relationship" explanations below).

The SKOS ontology is an ideal option for representing this type of knowledge (where concepts are represented as individuals/instances instead of classes) and we use it in the ISF to organize our vocabularies. The top SKOS class "Concept" is subclassed for a specific vocabulary and then class instances are created for each vocabulary entry. We are still developing this model, but we are currently using it for ICD9, MeSH, and UMLS codes (with appropriate SKOS relationships) and few other vocabularies being considered.

Links and references

[1] http://code.google.com/p/eagle-i/
[2] http://sourceforge.net/apps/mediawiki/vivo/index.php?title=Ontology
[3] https://www.eagle-i.net/
[4] http://vivoweb.org/
[5] http://www.ctsaconnect.org/
[6] http://www.ctsasharecenter.org/
[7] http://www.ifomis.org/bfo
[8] http://www.w3.org/2004/02/skos/
[9] http://www.w3.org/TR/owl2-overview/
[10] http://code.google.com/p/connect-isf/source/browse/
[11] http://purl.obofoundry.org/docs/index.html
[12] http://www.w3.org/TR/vcard-rdf/
[13] http://xmlns.com/foaf/spec/
[14] Courtot M, Gibson F, Lister A, et al: MIREOT: the Minimum Information to Reference an External Ontology Term. In Proc. ICBO'09 2009.
[15] https://github.com/CTSIatUCSF/plumage