Using Semantic Web in ICD-11: Three Years Down the Road

Tania Tudorache, Csongor I. Nyulas, Natalya F. Noy, and Mark A. Musen

Stanford Center for Biomedical Informatics Research, Stanford University, USA
{tudorache,nyulas,noy,musen}@stanford.edu

Abstract. The World Health Organization is using Semantic Web technologies in the development of the 11th revision of the International Classification of Diseases (ICD-11). Health officials use ICD in all United Nations member countries to compile basic health statistics, to monitor health-related spending, and to inform policy makers. In 2010, we published a paper in the ISWC In Use track reporting on our experience in the first six months with building and deploying iCAT, a Semantic Web platform to support the collaborative authoring of ICD-11. Three years since our original publication, 270 domain experts around the world have used iCAT to author more than 45,000 classes, to perform more than 260,000 changes, and to create more than 17,000 links to external medical terminologies. During the last three years, the collaboration processes, modeling and tooling have evolved significantly, and we have learned important lessons, which we will report in this paper. We describe the benefits of using semantic technologies as an infrastructure, which proved to be critical in making support for this rapid evolution possible. To our knowledge, this effort is the only real-world project supporting the collaborative authoring of ontologies at this scale, and which, at the same time, has a high visibility and impact for the health care around the world. We believe that the insights that we gained and the lessons that we learned after four years into this large-scale project will be useful to others who need to support similar collaborative projects.

1 Introduction

The World Health Organization (WHO) is leading a large international effort to develop the 11th revision of the International Classification of Diseases (ICD-11). ICD is the standard diagnostic classification for epidemiology, health management and clinical purposes. This standard classification has a tremendous international importance. United Nations member countries are using ICD to analyze the general health of population groups, to monitor the prevalence of diseases and other health problems, to compile mortality and morbidity statistics, to process insurance claims and to make decisions for resource allocation [21]. WHO publishes major revisions of ICD approximately every decade or more to ensure that the classification reflects the latest scientific status of the medical field. The 10th revision of ICD was published in 1990.

H. Alani et al. (Eds.): ISWC 2013, Part II, LNCS 8219, pp. 195–211, 2013.

ICD-11 is revolutionary compared to its predecessors: it uses Semantic Web technologies for the modeling of ICD, for the software infrastructure and for the support of collaboration processes. Our group has developed iCAT, a Web-based platform—built as an extension of our Protégé system—that WHO uses as the technological infrastructure for the development of ICD-11. In our initial report on the use of Semantic Web technologies in ICD-11 [14], we described our experience with the use of iCAT after it had been in production use for six months. Three years after that initial report, iCAT is in very active use and WHO intends to use it indefinitely for future major and minor revisions of ICD. Since our initial report, the user base, modeling, tooling and collaboration processes have evolved significantly in ways that we could not have predicted when we started the project. We have over 270 users who performed more than 260,000 changes, created more than 17,000 links to external ontologies, and posted over 61,000 comments. This increase is significant compared to our initially reported numbers, when we had 48 users, approximately 15,000 changes and 5,000 notes—a 6-times increase in changes, and 12-times increase in notes' counts.

During our ongoing engagement in this real-world project, we have learned a number of lessons, some of which were unexpected. One such lesson was that the use of Semantic Web technologies has brought us benefits internally, enabling us to build and maintain the collaborative platform as the requirements evolved drastically. As in many other projects, WHO could not articulate specific requirements at the beginning: The use of a Web-based collaborative platform was completely new for them and it was hard to predict which features will both enable and encourage contributions. We built the initial system the way that we thought it should be. Once the users and project managers have seen and used the initial system, they were able to articulate some of the requirements, which, however, changed frequently during the entire life cycle of the project. We learned that the use of Semantic Web technologies allowed us to be very agile in supporting the changing requirements in terms of the modeling, tooling and collaboration processes. In this paper, we report on the evolution of the project and on the benefits that the semantic technologies brought us. Specifically, this paper makes the following contributions:

- We describe the iCAT system that more than 270 medical experts around the world use to edit ICD-11. To the best of our knowledge, this effort is the largest collaborative ontology-development effort.
- We describe the lessons learned and how we had to evolve the system as the project moves through its life cycle stages to a mature state.
- We describe the benefits of using Semantic Web technologies not only for the well-known "external" benefits, such as making semantics explicit or enabling reuse, but also for the flexibility and agility that they provide in building and maintaining the system under frequently changing requirements.

The paper is organized as follows. Section 2 describes briefly the iCAT system. Section 3 provides usage statistics. In Section 4, we describe how the system has evolved in direct response to the requirements and usage of the platform. We then discuss the benefits of using semantic technologies in Section 5, and conclude in Section 6.

2 System Description

WHO is building ICD-11 as an ontology [14,12] represented in OWL [3] and iCAT[1] is a Web-based platform for authoring it. We had two main goals with implementing iCAT: (1) enable domain experts who are distributed around the world to edit the ICD ontology collaboratively; and (2) provide an interface that is easy enough to use and that will allow non-ontology experts to edit the ICD ontology.

Our group, in collaboration with WHO and the Health Informatics Modeling Topic Advisory Group (which is lead by Dr. Musen, one of the co-authors of this paper), created the core part of the ICD ontology. This core defines the main classes, properties and modeling patterns in the ontology. The domain experts only extend the core ontology with the domain content. The ICD core ontology also “drives” the user interface, as we explain later in this section.

The iCAT platform is a customization of WebProtégé [18,10]—a Web client for the Protégé[2] ontology editor [8]. Similarly to WebProtégé, iCAT is implemented in Java, it uses the Google Web Toolkit (GWT) and is open source.[3] iCAT takes advantage of many WebProtégé features, such as the concurrent browsing and editing of an ontology by distributed users, the read–write access policy mechanism, or the form-based editing, just to name a few. We described the architecture of the iCAT system in our previous work [14,13,15]. In this paper, we give a brief overview of iCAT and then focus on the new features that we have added since our earlier publications.

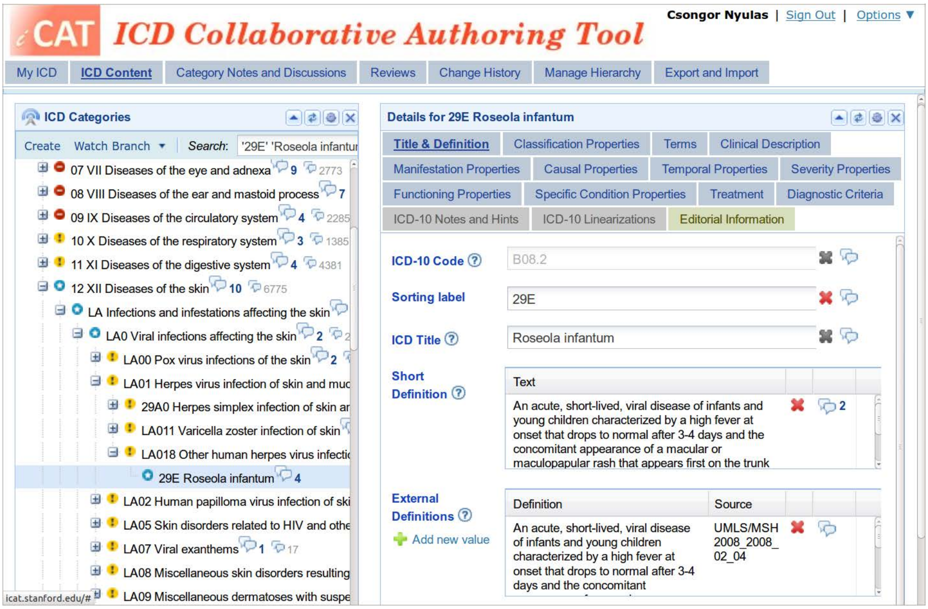
Medical experts use a form-based layout (Figure 1) to enter descriptions of diseases. Each tab in the interface (e.g., the *ICD Content*, *Change History*) provides features for a different task. The domain experts spend most of their time in the *ICD Content* tab where they enter information about diseases. The project managers make heavy use of the *Change History* tab when they curate the content. The *ICD Content* tab consists of a disease taxonomy (is-a hierarchy) and a forms interface. The forms display the properties of the selected disease.

The form-based interface associates an editing widget to a property in the underlying ontology. Users have to fill in 56 properties for a disease, out of which 52 properties are reified (i.e., the values are represented as instances). We represent many of the reified properties as tables, some as checkboxes or radio buttons. For example, the *External Definitions* widget in Figure 1 shows in a table the textual definitions for the selected disease coming from external terminologies; each row is an external definition represented as a reified instance;

[1] http://icat.stanford.edu

[2] http://protege.stanford.edu

[3] Source code available in the SVN repository at:
https://smi-protege.stanford.edu/repos/protege/icat/trunk


iCAT ICD Collaborative Authoring Tool
Csongor Nyulas | Sign Out | Options ▼
My ICD
ICD Content
Category Notes and Discussions
Reviews
Change History
Manage Hierarchy
Export and Import
ICD Categories
Create
Watch Branch ▼
Search:
'29E' 'Roseola infantu
07 VII Diseases of the eye and adnexa 9 2773
08 VIII Diseases of the ear and mastoid process 7
09 IX Diseases of the circulatory system 4 2285
10 X Diseases of the respiratory system 3 1385
11 XI Diseases of the digestive system 4 4381
12 XII Diseases of the skin 10 6775
LA Infections and infestations affecting the skin
LA0 Viral infections affecting the skin 2 2
LA00 Pox virus infections of the skin 2
LA01 Herpes virus infection of skin and muc
29A0 Herpes simplex infection of skin ar
LA011 Varicella zoster infection of skin
LA018 Other human herpes virus infectio
29E Roseola infantum 4
LA02 Human papilloma virus infection of ski
LA05 Skin disorders related to HIV and othe
LA07 Viral exanthems 1 17
LA08 Miscellaneous skin disorders resulting
LA09 Miscellaneous dermatoses with suspe
icat.stanford.edu/#
Details for 29E Roseola infantum
Title & Definition
Classification Properties
Terms
Clinical Description
Manifestation Properties
Causal Properties
Temporal Properties
Severity Properties
Functioning Properties
Specific Condition Properties
Treatment
Diagnostic Criteria
ICD-10 Notes and Hints
ICD-10 Linearizations
Editorial Information
Add new value
2

ICD-10 Code: B08.2

Sorting label: 29E

ICD Title: Roseola infantum

Short Definition:

Text
An acute, short-lived, viral disease of infants and young children characterized by a high fever at onset that drops to normal after 3-4 days and the concomitant appearance of a macular or maculopapular rash that appears first on the trunk

External Definitions:

Definition	Source
An acute, short-lived, viral disease of infants and young children characterized by a high fever at onset that drops to normal after 3-4 days and the concomitant	UMLS/MSH 2008_2008_02_04

Fig. 1. The iCAT system used by medical experts around the world to author ICD-11. Each tab in the interface offers particular functionalities. For example, the *ICD Content* tab, which is currently selected, provides the main editing features. The panel on the left shows the disease class tree with icons indicating the editorial status of classes, as well as the number of notes in that particular ontology branch. The panel on the right presents a tabbed interface for different property categories that describe a disease. The current tab shows the title and definitions of the selected disease.

the *Definition* and *Source* columns represent properties of the reified instance. This form presentation gives us a way to hide the underlying OWL representation from the users, and to present them with a simple and familiar editing interface that they can understand.

Another important feature of iCAT is its ability to create links [15,17] between properties of diseases and external terminologies stored in the BioPortal ontology repository [20]. Users can search external terminologies directly from the iCAT interface and then import references to terms from these terminologies. These references are represented in the ontology as reified instances that store the term reference metadata. About two thirds of the reified properties for a disease (31 out of 52) are references to external terminologies. We partition the properties of diseases into different tabs in the interface based on the class' `rdf:types`. For example, there is a tab for title and definition properties, a tab for the clinical description of a disease, and a tab for diagnostic criteria. Thus, a user can focus easily on a set of properties that are of interest to her. Some class branches in ICD, such as the *External causes of morbidity and mortality*, have different properties associated with them, and the user interface configures automatically based on the class types.

From its first release, iCAT provided significant support for collaboration. As the project matured, we had to add new features in direct response to the requirements of the users and changes in the workflow. Distributed users edit the ontology simultaneously, and iCAT propagates the changes to all client Web browsers immediately, thus minimizing the risk of conflicting changes. iCAT tracks all the edits a user makes and stores these changes as instances in a Change and Annotation Ontology (ChAO) [7], creating a structured log that we can easily access programmatically. Users can carry out threaded discussions using structured notes. The types and structure of the notes are also defined in ChAO. A user can attach a note to a disease (the icons next to the class name in the tree from Figure 1), or to a triple. For example, in Figure 1, there are four notes attached to the disease as shown in the class tree on the left hand side, and there are two notes attached to the *Short definition* of the selected class shown on the right hand side. Users can watch entities or branches in the ontology and receive email notifications about the changes and notes in their watched areas.

To provide access control in iCAT, we represent users, groups, operations and access policies in a lightweight ontology [16,9]. Using an ontology to specify access-control mechanism gives us significant flexibility in extending this mechanism as the project evolves. Besides the read and write permissions, iCAT has more granular access control at the level of operations (e.g., create class or move in hierarchy), or even at the level of a property (e.g., User X cannot edit the *Short definition* property of a disease).

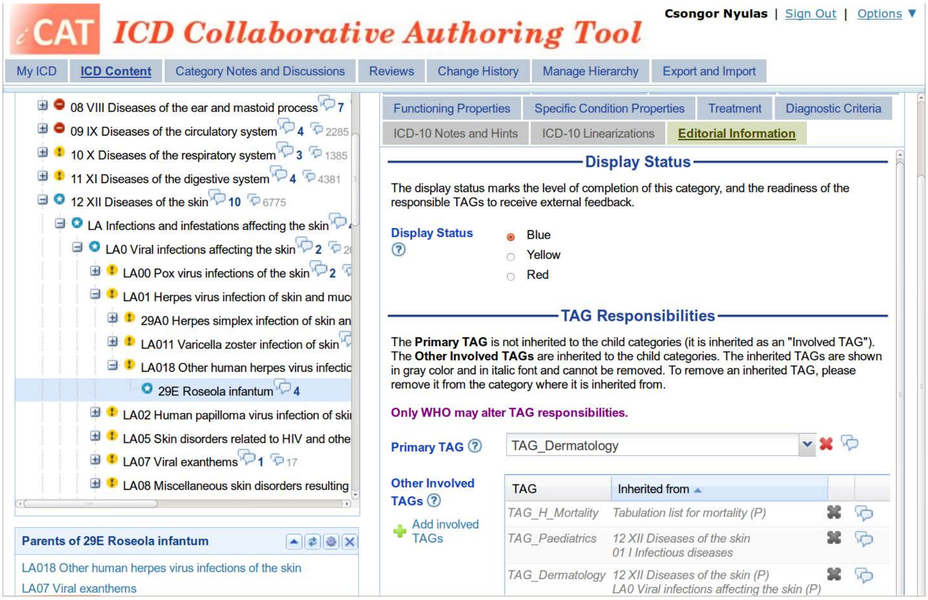
As the result of the project evolution, we implemented editorial support in the *Editorial Information* tab (Figure 2): For example, classes are assigned a three-level *display status* that marks the level of maturity of the class definition and the fraction of property values that are filled in. Each class is also assigned a group of users who are “in charge” of that class and its subclasses.

3 iCAT in Use

iCAT is running in a production setting since November 2009. Different types of users are using the platform in their daily work: classification experts are mainly interested in the organization of diseases in parent–child hierarchies, as well as with the consistency of the different views that will be extracted from ICD; domain experts are mainly medical doctors who use the platform to enter properties of diseases (for example, definitions, synonyms, clinical descriptions, manifestations); project managers oversee the project and curate the content.

iCAT currently has 273 registered users out of which 102 have made changes. There are 45 user groups (one user might belong to different groups) reflecting different administrative organization of the project. Some groups are formed based on access permissions.

Figure 3 shows the cumulative number of changes and notes made in the system. Users have made 263,628 changes and have created 61,589 notes. The most prolific user has made 72,056 changes, while the average number of changes per user is 2,511, and the median is 84. The most “talkative” user has created


iCAT ICD Collaborative Authoring Tool
Csongor Nyulas | Sign Out | Options ▼
My ICD
ICD Content
Category Notes and Discussions
Reviews
Change History
Manage Hierarchy
Export and Import
08 VIII Diseases of the ear and mastoid process 7
09 IX Diseases of the circulatory system 4 2285
10 X Diseases of the respiratory system 3 1385
11 XI Diseases of the digestive system 4 4381
12 XII Diseases of the skin 10 6775
LA Infections and infestations affecting the skin
LA0 Viral infections affecting the skin 2
LA00 Pox virus infections of the skin 2
LA01 Herpes virus infection of skin and muc
29A0 Herpes simplex infection of skin an
LA011 Varicella zoster infection of skin
LA018 Other human herpes virus infectic
29E Roseola infantum 4
LA02 Human papilloma virus infection of ski
LA05 Skin disorders related to HIV and othe
LA07 Viral exanthems 1 17
LA08 Miscellaneous skin disorders resulting
Parents of 29E Roseola infantum
LA018 Other human herpes virus infections of the skin
LA07 Viral exanthems
Functioning Properties
Specific Condition Properties
Treatment
Diagnostic Criteria
ICD-10 Notes and Hints
ICD-10 Linearizations
Editorial Information
Add involved TAGs

Display Status

The display status marks the level of completion of this category, and the readiness of the responsible TAGs to receive external feedback.

Display Status ⓘ ● Blue ○ Yellow ○ Red

TAG Responsibilities

The **Primary TAG** is not inherited to the child categories (it is inherited as an "Involved TAG"). The **Other Involved TAGs** are inherited to the child categories. The inherited TAGs are shown in gray color and in italic font and cannot be removed. To remove an inherited TAG, please remove it from the category where it is inherited from.

Only WHO may alter TAG responsibilities.

Primary TAG ⓘ: TAG_Dermatology

Other Involved TAGs ⓘ

TAG	Inherited from ▲
TAG_H_Mortality	*Tabulation list for mortality (P)*
TAG_Paediatrics	*12 XII Diseases of the skin* *01 I Infectious diseases*
TAG_Dermatology	*12 XII Diseases of the skin (P)* *LA0 Viral infections affecting the skin (P)*

Fig. 2. The *Editorial Information* tab in iCAT allows project managers to track the level of maturity of a disease class, and to assign groups who are responsible for a class and its children. The *Primary TAG* denotes the group who is primarily responsible for this branch of the ontology, the *Other Involved TAGs* have also an interest in this branch, and are inherited to descendent classes.

16,949 notes, while the average number of notes per user is 978 and the median is 35. As we mentioned in Section 2, the notes have different types. When users create a note, they can choose the type of the note. The most common note type is *Comment* with 45,753 occurrences (it is also the default selection), followed by *Explanation* and *Reference*. Two thirds of the created notes are attached directly to classes, while one third is attached at the level of a triple.

Since November 2009, the ICD ontology has grown to 45,028 classes out of which 16,204 have textual definitions. The number of definitions are a good indicator for classes that are in a "mature" state: content experts have reviewed and updated the class and it is now ready for external feedback. Users have added 112,496 index terms for an electronic index for looking up ICD diseases. There are 17,706 references to external biomedical ontologies and terminologies.

These numbers demonstrate that the users are actively engaging with the iCAT system both to create content and to collaborate. We have performed several usability studies on the system and we report them elsewhere [14,13]. The focus of this paper is not the iCAT system itself, but rather our lessons learned and benefits from using semantic technologies in building the system.

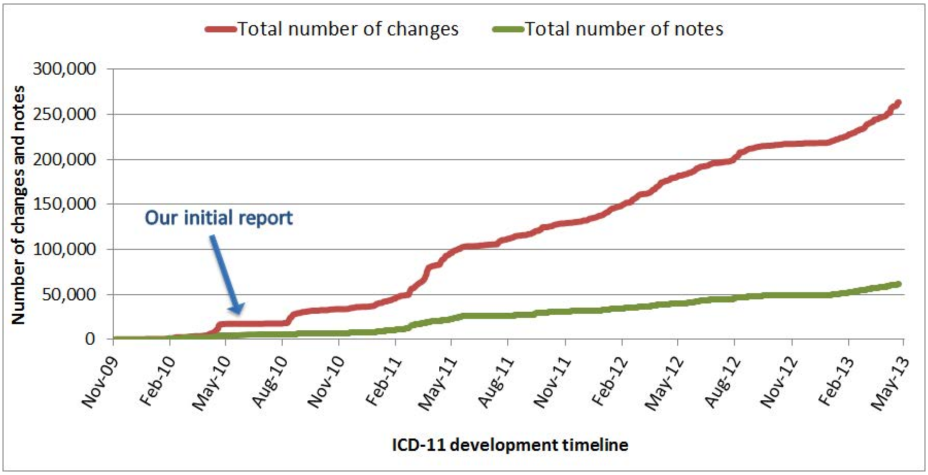


Fig. 3. The cumulative number of changes and notes in iCAT since its deployment in November 2009. Our initial report on the system used the data in June 2010 (marked with the arrow in the graph).

4 Evolution of the System

Section 3 described the current state of iCAT—a fairly complex and mature system. We started with a much simpler system in 2009. Over time, as larger group of collaborators started using the system and the project evolved through different life stages towards a mature state, WHO and other participants in the ICD-11 provided a continuous stream of new requirements. We were able to address these new requirements in an agile way because our infrastructure used semantic technologies that provided significant flexibility and power (see Section 5).

In this section, we describe the evolution of the system over the past three years and discuss which semantic technologies enabled us to support this rapid evolution. We focus on three aspects of iCAT evolution: (1) evolution in terms of modeling; (2) evolution of iCAT features; and (3) evolution of the collaboration processes that we had to support. While some of these changing requirement are specific to our project, many are likely to occur in the large-scale deployment of any collaborative knowledge-intensive application.

4.1 Evolution of Modeling

Previous revisions of ICD had very little structure in them. The classification used to be a long list of diseases with their associated codes, and a few properties, such as synonyms, inclusions and exclusions. The diseases did not even have a textual definition associated to them. When we started the project, ICD-10 was published in XML format using the Classification Markup Language (ClaML) [6]

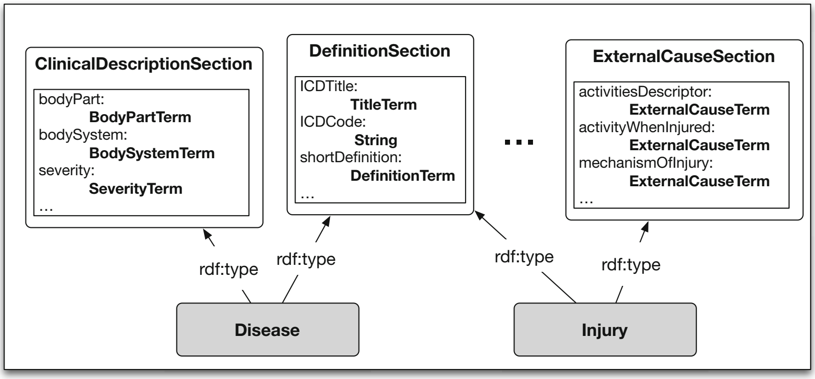


Fig. 4. The use of metaclasses to categorize different types of domain entities, for example, diseases and injuries. The `owl:Class` *Disease* has several metaclasses as its `rdf:type`. Each metaclass defines the properties that appear on a single tab in the editing interface (see Figure 1). Different types of entities will have different metaclasses (e.g., *Disease* and *Injury* share some of the metaclasses, but the *ExternalCauseSection* only applies to the *Injury* class).

as a schema. The initial ICD-11 ontology simply modeled the ClaML schema, and we wrote an importer and exporter from and to the ClaML files to generate the first version of the ICD ontology.

Content model and metaclasses. WHO has developed a *content model* for ICD that defines properties describing diseases and related entities. The initial content model was simple: Each disease contained textual definitions and fewer than 10 other properties. These properties supported different use cases of the classification. Over time, however, the domain experts wanted to differentiate between *types* of entities (e.g., diseases, injuries, external causes), which were described by different sets of properties. For example, an *injury* had properties describing the mechanism of injury and the place of occurrence, while a *disease* did not have these properties. At the same time, the domain experts wanted to add more and more information to ICD classes so that ICD could serve a larger number of use cases. As the result, the current system uses 56 properties that the user needs to fill in for a disease. We used *metaclasses* to categorize and group different properties of the main entities in the domain (Figure 4). *Metaclasses* are OWL classes whose instances are themselves classes. A metaclass defines the properties of the class itself. For example, one metaclass defined the properties describing the definitions of a disease, another metaclass contained the properties describing the clinical description of a disease, and so on. A disease class is an instance of these different metaclasses. The initial ontology contained 3 metaclasses. Currently, we use 12 metaclasses to describe ICD classes. As the number of properties and metaclasses grew, we needed a flexible way of adapting the iCAT user interface without changing the code, sometimes even on the fly without restarting the system. Furthermore, certain branches in ICD, such as the *External causes of morbidity and mortality*, have completely different properties describing them (e.g., *Intent*, *Mechanism of Injury*, *Object/Substance*), and therefore different metaclases and a different form in the interface.

Views. A major development during the first year of the project was the decision by WHO to extract different views out of the ICD ontology for specific use cases. Each view contains a subset of classes and properties and conforms to the principles of statistical classifications (e.g., single parent hierarchies) [2,14]. Users were able to specify views for morbidity, mortality, primary care, low resource settings, and so on. Thus we had to provide the modeling constructs and the user interface that would support the definition of such views. We modeled the views as another set of properties for a disease that describe which views the disease belongs to and a set of properties of the inclusion in the view. The *Classification Properties* tab in the main editing screen (Figure 1) enables users to specify this information. The major modeling effort for representing the views and the extension of the user interface to support it had to happen while the iCAT system was in production use and had to be compatible with the previous modeling and content.

Backwards compatibility. A similar major modeling effort happened when the project leaders decided to make explicit the backwards compatibility of ICD-11 with previous revisions of the classification. We had to make again a major modeling overhaul to represent ICD-10 and its different variants for different countries and to make a mapping to the current ICD-11 classes. Naturally, this modeling and mapping effort would have been much smaller, if it would have happened at the beginning of the process. After debating different alternatives for modeling the backwards compatibility, we were able to reuse the views mechanism that we mentioned earlier for this purpose: essentially, a mapping to ICD-10 was another view on ICD-11.

Post-coordination. Perhaps the biggest modeling overhaul so far is happening at the time of this writing, four years since the beginning of the project. We now need to incorporate in the model the possibility to *post-coordinate* certain properties of a disease so that not all possible combination of properties are encoded in the classification. For example, a disease has a *severity* property, which in some cases can be *mild, moderate* or *severe.* Other properties, such as *temporality* can have values as *acute* or *chronic.* Currently, each possible combination results in the definition of a new OWL class in ICD (e.g., there may classes such as *Mild acute pancreatitis* and *Severe acute pancreatitis*). Rather than creating all possible combinations of these properties, we need to be able to specify which properties can be post-coordinated, and the rules for the post-coordination. Currently, there are 21 properties describing a disease that can be post-coordinated. This modeling effort is enormous and still on-going, and will have deep implications on the presentation of the ontology, the extraction of the views and the user interface.

4.2 Evolution of the Features

As the project evolved, we addressed requirements for many new features, large number of which were generic. We now focus on the evolution of the features that we think could apply to other collaborative projects.

Avoiding duplicates. As more people are editing within the system, and the class hierarchy becomes deeper and more classes have multiple parents, users started creating duplicate classes with similar names, because they were not aware that the class already existed. This situation naturally arises in a collaborative project, in particular one where user groups overlap in their coverage of the ontology. Therefore, we implemented support for autocomplete and enhanced the class-creation mechanism to search for similar class names while the user typed the name of the new class.

External access. As the project evolved, we implemented a series of requested sharing and access features. For example, it became critical to have each class accessible via a unique URL, so that users can send links to their collaborators or to specify the location of a problem in a bug report. We also use the class-specific URLs in email notifications about changes in the ontology. As the project got more traction, external groups requested the ability to access the content programmatically. In response, we created nightly dumps of the ICD ontology and provided a Java API to access it [1]. This access enabled WHO itself to implement a public browser for ICD [22]. This browser reads the nightly dumps and uses the Java API to interpret it and to present a read-only version of the classification to the world. Anyone can browse the classification, and add notes and make proposals for changes. WHO has also implemented a Web service that provides the content of ICD classes as JSON or XML.

Export to spreadsheet. Another important feature that came out while the project was unfolding was the ability to export the content to spreadsheets. Users did not want to feel “locked-in” to using a Web-based system and wanted to be able to share the content with collaborators who were not part of the iCAT system, and to use a familiar spreadsheet application. We worked with a few experienced users to create Excel templates using macros. These templates not only provide a way to browse the content, but also enable edits. The spreadsheet export contained a branch of the ontology, including the taxonomy, and a part of the properties attached to a disease. The export feature is very popular and the domain experts invoke it very frequently. Even though the domain experts prefer to edit in familiar tools such as Excel, this tool can only account for a small percentage of the actual model of a disease, as the ICD ontology had a fairly complex structure that is hard to map to a tabular format.

ID scheme. When we started the project, there was no ID scheme in place for naming the ICD entities. We created an internal naming scheme that used Universally Unique Identifiers (UUID). We used a WHO namespace together with these 128-bit numbers to create unique identifiers for ICD entities. During the past year, the groups involved in the project engaged in a vigorous discussion on which naming scheme to use as the "public and official" identifiers for ICD. In the end, we agreed on a single public ID naming scheme. This public ID is a resolvable URL that will return the details of the class. However, this URL is different from the internal IDs that we already used throughout iCAT. Other groups have also already relied on the existing IDs to make mappings, or for other purposes. Therefore, we decided to keep the internal IDs and to store the public ID as a datatype property on each ICD entity, so that both are accessible to the different users.

4.3 Evolution of the Collaboration Processes

iCAT started as an open platform, it did not require a log in, anyone could browse the content of ICD as it was authored. Less than a year into the project, the groups participating in the process requested that we close the access to the platform, allowing only registered users (registration is moderated) to access the classification-in-the-making. The request was motivated by early negative reactions from the public to the classes that were still under development. These reactions only amplified the feelings that many domain experts already had about making public a classification that was under active development. Furthermore, the change tracking recorded all the changes in the system using identifiable user names, providing yet another incentive not to make the system public. Even though in our planning discussions, everyone assumed we will use an open platform, the pragmatics of the process made us change our plans.

The activity in iCAT has varied a lot across different groups and different users in the group, and we needed a way to incentivize users to become more active. One way to motivate users to contribute more was to displays a Top 10 of the "most active" ontology branches in ICD (which were usually associated to certain working groups). We had reports from the users who really cared if their assigned branch was visible in the Top 10 or not. We are also displaying two graphs with the changes and notes showing the activity of each user. We were able to provide these kind of statistics and more detailed ones by using the structured logs stored in the Changes and Annotation Ontology (ChAO).

As the project grew larger, WHO wanted to have an overview of the process in order to keep track of the progress of the project. Using the information in ChAO, we implemented a series of project-management plugins [4] that present different views and statistics of the process. Our collaborators have also implemented a graphical tool, PragmatiX [19], that allows project managers to get a quick understanding of the process, and provides several meaningful aggregations of the changes and notes information.

We started the project supporting a fairly simple access policy mechanism enabling read and write privileges at the level of the ontology. During the project, the requirements have changed, and we implemented more granular access. For example, certain users kept making wrong changes to certain properties. Or, only WHO should be able to change certain properties (for example, marking a class as "obsolete"). Therefore, we had to extend our access policy mechanism to support policies at the level of a property in the ontology. Later in the process, WHO wanted to restrict access to certain operation, such as class creation, move in the hierarchy, or changing the definition of views. Our access policy mechanisms was once more extended to provide support for these features. Once the access to certain operations was blocked, we were asked to provide "temporary passes" for unrestricted access to certain users to allow them to finish their work. We were able to support all these features because of flexibility we had from representing our access policies using an ontology.

4.4 Evolution: Summary of Lessons Learned

The past four years accounted for one of the first long-term experiences in using a semantic-web application in a large-scale international collaborative project. One of the key lessons to take away from the experience is that requirements about every aspect of the project change constantly and the system grows more and more complex in its functionality. This lesson of course parallels the development of large software projects everywhere. The rapid change in requirements in our case was likely exacerbated by the sheer novelty of the application in this domain. As we discuss in the following section, it was the use of semantic technologies internally in the system for many aspects of the system—from specification of metaclasses, to the use of ontologies to represent changes and access policies, to the use of semantics to describe views, and so on—enabled us to adjust the system quickly. In many cases we did not need to change the software or even to shut down the system while performing major modeling overhauls.

5 Benefits of Using Semantic Web Technologies

When we started using Semantic Web standards and technologies in this project, we did expect that we will have the "traditional" benefits, which we will refer to as *external benefits* throughout this section. We did not expect, however, that we, as the implementers of a Semantic Web system, will also benefit internally as we build and maintain the system. We call the latter benefits *internal*.

5.1 External Benefits

Using OWL, a W3C standard, to represent ICD-11 already comes with a series of advantages. First, by using this standard we can rely on well-defined semantics, which ensures that all tools and users will interpret it in a uniform way. The previous approach, which used ClaML, came with a clear way of defining the data structures in a DTD, but the interpretation of the constructs were not formally defined, leaving room for different interpretation by different tools. Second, using OWL opens the door of reusing ICD in other biomedical ontologies, or for other purposes in applications that need machine-processable information. Third, ICD in OWL naturally has unique identifiers for all its entities (not just for the classes), a critical feature in allowing its reuse in other applications. Because ICD is a standard classification that is used throughout the world, making it more Web and machine-friendly will not only bolster its uptake in coding systems but also will open it up for unprecedented uses. Forth, using the same standard representation as other large medical terminologies that chose to use OWL, will make it easier for ICD to create links or mappings between these ontologies, thus encouraging interoperability of biomedical tools. Fifth, because ICD-11 is an OWL ontology, it can be uploaded to the BioPortal repository, which will make the plethora of BioPortal features available. For example, BioPortal users can include ICD-11 in automatic annotation of resources, or they may access the ICD content through the BioPortal REST services and through the BioPortal SPARQL endpoint [20].

We used many OWL language features in the ICD representation: class–subclass relations, subproperties, domains and ranges, cardinalities, inverse properties, equivalent classes, to name a few. The OWL expressivity of the ICD ontology is SHOIN(D). We can now use a DL reasoner to check the consistency of our ontology. We had cases in which we discovered that the ontology was inconsistent, and we could use the explanation support [5] to pinpoint the statement that caused the inconsistency. Without a formal representation and such tools, finding the inconsistency would have been very difficult. We plan to use a DL reasoner to debug and fix the manually created poly-hierarchies. Once we implement the support for post-coordination in ICD, we also plan to use DL reasoning to check and debug the post-coordination definitions.

5.2 Internal Benefits

We refer to the benefits that helped our team directly with building and maintaining the tool itself as internal benefits. iCAT is a customization of the more generic WebProtégé, and we could take advantage of the generality of the tool combined with the semantic modeling. This approach provided the flexibility that we needed in order to deal with the frequently changing requirements.

The iCAT interface is a domain-expert–friendly rendering of the underlying OWL ontology. As we mentioned in Section 4, the core OWL ontology kept changing frequently even after iCAT has gone into production. We had to find a way to **adapt the user interface** with minimal or no changes to the code in response to the core ontology changes. These changes included the addition of new properties to disease description, definition of new metaclasses, and customization of properties for different branches in the ontologies. There were several parts to our solution. First, we built the user interface forms in a generic way, in which we bind a widget (e.g., a table) to a property in the ontology in a declarative way. Second, we used as much as possible the same OWL modeling patterns for different properties in the ontology (e.g., the reified properties), so that we can easily reuse the same widgets for different properties. Third, we created a declarative representation [11] of the user interface that made the binding between the user interface widgets and the underlying ontology properties explicit. We also made this declarative representation extensible, as we anticipated that further customizations of the widgets will be necessary in the future.

We generate the iCAT user interface dynamically from the ontology and the declarative layout representation. Therefore, changes in the ontology result in changes in the user interface immediately, without the need to change the iCAT code or restart the system. This design allowed us to easily adapt the user interface whenever the underlying core ontology changed, which happened frequently: the ontology started with about 10 properties describing a disease, and currently there are 56 (see Section 4). Similarly, we started with only 3 metaclasses that defined what properties users see for each disease (Figure 4), but now have 12 such metaclasses.

Furthermore, ICD has other types of classes that are not diseases, but rather injuries or external causes. These special types of classes are part of the *External causes of mortality and morbidity* branch in the ontology. The classes in this branch have different metaclasses defined that only apply for this branch, and do not have other metaclasses that apply to the "normal" disease, such as *ClinicalDescription*. To address this new requirement, we have extended the generic declarative forms mechanism to encode the types for which a particular tab and property should show up in the user interface. For example, we have defined a rule that the *Clinical Description* tab should show up in the interface only if the selected class also has the metaclass *ClinicalDescription* as its type. In general, the dynamic generation of the user interface allowed us to present different components for different types of classes and properties based on the semantic content of the class.

The generic user interface and declarative binding between the user interface and the ontology allowed us to **reuse** iCAT as a production platform for two other WHO classifications: the International Classification of Traditional Medicine (ICTM) and the International Classification of Patient Safety (ICPS). To support the two classifications, we did not have to change the iCAT code almost at all. We built only two custom widgets for ICPS. For the rest of the customization, all we had to do was to build the core ICTM and ICPS ontologies and to create the declarative user interface bindings.

We had similar benefits by representing the **access policies** as a lightweight ontology [9,16] that we could easily extend. The users, groups and projects are represented as instances in this ontology. The access policies (e.g., *The WHO group is allowed to read and write the ICD ontology*) are represented as instances that associate a group with a set of allowable operations for that group and a particular project. However, as the project evolved, we had to create new types of operations, which we could easily add directly to the ontology by creating additional instances of the class *Operation*. For example, we created instances for *Create class* or *Move in hierarchy*. Then we added new access policies to the project instance (e.g., *Only WHO group is allowed to create class and move in hierarchy in the ICD ontology*). The ontology gives us a flexible way of representing the access policies in a declarative way, without needing to change the access policy code at all. We did, however, need to add support in the user interface code for enforcing the additional policies. This additional development proved to be minimal. Once we had the access-permission enforcement implemented, it was very easy for us to support different workflows for WHO. For example, when they requested us to create *temporary passes* (see Section 4) for certain users, we could do this by editing the access policy ontology, and the changes took effect in real time.

We have also used a **semantic representation for the changes and notes** using the ChAO ontology [7]. The ChAO ontology provided us, as in the previous two examples, with a declarative representation of the change logs and of the threaded discussions. Furthermore, the ChAO dictates the taxonomy and structure of the notes used in the platform. We found two main benefits of using an ontology for representing changes and notes. First, we were able to add new note types on the fly in the ChAO ontology, and the user interface would pick it up right away. For example, at some point, WHO asked us to add a new note type, *Reference* where users should encode the evidence in the scientific literature for the different statements in the ICD ontology (e.g., the source of a disease definition). Having different types of notes, and being able to "just" declare a new type provided us a lot of flexibility in terms of supporting collaboration. Second, because we record changes and notes in a structured and formal way, we are able to perform intelligent analyses of ontology evolution using different algorithm and different visualizations. We can then present these analyses to the users and to the project managers. Using the structured logs and the typology of changes, we were able to identify emerging roles of the users by using statistical analysis [4]. We were also able to present meaningful visualizations of the current state of the ontology to the project managers and enable them to make informed decisions [19,4].

6 Conclusion

We presented our long-term experience in implementing and supporting a Semantic Web application in a large-scale international collaborative project. We described the Semantic Web platform, iCAT, that is used by over 270 medical domain expert in the world to author the ICD-11 ontology. One of the key lessons to take away from the experience is that requirements about every aspect of the project change constantly, and the system grows more and more complex in its functionality. We found significant benefits in using semantic technologies not only for their external benefits, which are now widely accepted, but also internally, as they provided us with the flexibility to adjust to changing requirements and to support a changing processes in an agile way. "Eating our own dog food" proved to be extremely useful, often in ways that we would have not predicted. The result is one of the most high-profile uses of Semantic Web in the real world.

Acknowledgments. We are very grateful to Timothy Redmond, Jennifer Vendetti, and Martin O'Connor for their help with the implementation of WebProtégé, and to Bedirhan Üstün, Robert Jakob, Can Çelik, Nenad Kostanjsek, Molly Meri Robinson, and Sara Cottler from WHO for providing the requirements for the project and their important feedback on iCAT. We would like to thank Samson Tu, Alan Rector and all the members of the HIM-TAG and RSG for their invaluable work in designing the ICD content model. This work was supported in part by grants GM086587 and GM103316 from the US National Institutes of Health.

References

1. Accessing the iCAT Content Programatically, https://sites.google.com/site/icatusers/home/api-access (last accessed: May 2013)
2. Bailey, K.D.: Typologies and taxonomies: An introduction to classification techniques. Sage Publications, Inc., Thousand Oaks (1994)
3. Dean, M., Schreiber, G.: Web ontology language (OWL) reference, http://www.w3.org/TR/owl-ref/ (last accessed: May 2013)
4. Falconer, S., Tudorache, T., Noy, N.F.: An analysis of collaborative patterns in large-scale ontology development projects. In: K-CAP 2011, vol. 11, pp. 25–32 (2011)
5. Horridge, M., Bail, S., Parsia, B., Sattler, U.: The cognitive complexity of OWL justifications. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 241–256. Springer, Heidelberg (2011)
6. ISO 13120:2013. Syntax to represent the content of healthcare classification systems – Classification Markup Language (ClaML), http://tinyurl.com/l5qs38d
7. Noy, N.F., Chugh, A., Liu, W., Musen, M.A.: A framework for ontology evolution in collaborative environments. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 544–558. Springer, Heidelberg (2006)
8. Noy, N.F., et al.: Creating semantic web contents with Protege-2000. IEEE Intelligent Systems 16(2), 60–71 (2001)
9. BMIR. Configuring access policies in Protege using the metaproject, http://tinyurl.com/ms6ut4h (last accessed: May 2013)
10. BMIR. WebProtègè, http://protegewiki.stanford.edu/wiki/WebProtege (last accessed: May 2013)
11. BMIR. WebProtege layout configuration, http://protegewiki.stanford.edu/wiki/WebProtegeLayoutConfig (last accessed: May 2013)
12. Tu, S.W., et al.: A content model for the ICD-11 revision. Technical Report BMIR-2010-1405, Stanford Center for Biomedical Informatics Research (2010)
13. Tudorache, T., Falconer, S., Noy, N.F., Nyulas, C., Üstün, T.B., Storey, M.-A., Musen, M.A.: Ontology Development for the Masses: Creating ICD-11 in WebProtege. In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS (LNAI), vol. 6317, pp. 74–89. Springer, Heidelberg (2010)
14. Tudorache, T., Falconer, S., Nyulas, C., Noy, N.F., Musen, M.A.: Will Semantic Web Technologies Work for the Development of ICD-11? In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part II. LNCS, vol. 6497, pp. 257–272. Springer, Heidelberg (2010)
15. Tudorache, T., Falconer, S., Nyulas, C., Storey, M., Ustun, T., Musen, M.: Supporting the collaborative authoring of ICD-11 with Webprotégé. In: AMIA Annual Symposium Proceedings, vol. 2010, p. 802. AIMIA (2010)

16. Tudorache, T., Musen, M.A.: Collaborative development of large-scale biomedical ontologies. In: Collaborative Computational Technologies for Biomedical Research, pp. 179–200 (2011)
17. Tudorache, T., et al.: Use cases for the interoperation between an ontology repository and an ontology editor. In: Proc. of SERES 2010, Shanghai, China (2010)
18. Tudorache, T., et al.: WebProtege: a collaborative ontology editor and knowledge acquisition tool for the web. Semantic Web 4(1), 89–99 (2013)
19. Walk, S., et al.: Pragmatix: An interactive tool for visualizing the creation process behind collaboratively engineered ontologies. IJSWIS (to appear, 2013)
20. Whetzel, P.L., et al.: Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucleic Acids Research 39(suppl. 2), W541–W545 (2011)
21. World Health Organization. International Classification of Diseases (ICD), `http://www.who.int/classifications/icd/` (last accessed: May 2013)
22. World Health Organization. The Public ICD Browser, `http://apps.who.int/classifications/icd11/browse/f/en` (last accessed: May 2013)