Structured Data Acquisition with Ontology-Based Web Forms

Rafael S. Gonçalves*, Samson W. Tu, Csongor I. Nyulas,
Michael J. Tierney and Mark A. Musen

Stanford Center for Biomedical Informatics Research
Stanford University, Stanford, California, USA

ABSTRACT

Structured data acquisition is a common, challenging task that is widely performed in the field of biomedicine. However, in some biomedical fields, such as clinical functional assessment, little effort has been done to structure functional assessment data in such a way that it can be automatically employed in decision making (e.g., determining eligibility for disability benefits) based on conclusions derived from acquired data (e.g., assessment of impaired motor function). In order to be able to apply such automatisms, we need data structured in a way that can be exploited by automated deduction systems, for instance, in the Web Ontology Language (OWL); the *de facto* ontology language for the Web. The rise of OWL caused a paradigm shift in knowledge systems from frame-based to axiom-based. Because of the axiom-based nature of OWL, it is more difficult to acquire instance data based on OWL than it was based on frames. In this paper we tackle the problem of generating Web forms from OWL ontologies, and aggregating input gathered through these forms as an ontology of "semantically-enriched" form data that can be queried using an RDF query language, such as SPARQL. The ontology-based structured data acquisition framework that we have developed is presented through its specific application to the clinical functional assessment domain, with examples of how one can perform desirable analyses of gathered data with simple queries.

1 INTRODUCTION

Ontology-based form generation and structured data acquisition was first pioneered almost 30 years ago. In the early 1990s, Protégé-Frames used definitions of classes in an ontology to generate knowledge-acquisition forms, which could be used to acquire instances of the classes [2, 3]. With OWL as the preferred modeling language for ontologies, class definitions are collections of description logic (DL) axioms, and can no longer be seen as templates for forms [9]. Unlike template-based knowledge representations, where what can be said about a class is defined by the slots of the class template, axiom-based representations do not have this kind of locally scoped specification, and allow any axiom describing the same class to be added to the ontology, as long as the axiom does not lead to inconsistencies. Template-based knowledge representation systems use closed-world reasoning and have local constraints (e.g., cardinality of a slot for a particular class) that can be validated easily, while in an axiom-based system with the open-world assumption such local constraint checking is much more problematic. Furthermore, in our chosen application domain, assessment instruments have specific formats that do not lend themselves to be seen as representing instances of domain ontology classes. Items in the instruments have potentially complex descriptions of information to be collected, such as the severity of pain with a particular quality, and at a specific anatomical location. The challenge is to model the assessment instruments and relate the assessed data to a domain ontology with which one can formulate meaningful queries.

In this paper, we describe a solution for representing, acquiring and querying assessment data that uses (1) domain ontologies and standard terminologies to give formal descriptions of entities in our chosen domain, (2) an information model of assessment instruments to drive the generation of data-acquisition Web forms, and (3) a data model for the acquired information that links the data to the domain ontologies and standard terminologies. Such linkage makes it possible to query and aggregate the data using the logical representation of the domain concepts in the ontologies.

2 RELATED WORK

In addition to the comparison with Protégé-Frames' template-based instance acquisition method described in Section 1, we briefly contrast our work with two other systems that are designed to use forms for acquiring structured data: the first targets the domain of patient assessment, which is similar to the work reported here, while the second is a generic Web-based technology from which one can draw examples on how to arrive at a domain-independent solution.

The clinical documentation system described in [6] uses a template schema to allow a technology-savvy clinician to create documentation templates that include the local structure of subforms and potentially complex clinical descriptions consisting of features and their values. The features and values are mapped to a medical ontology, and the system automatically generates ontological descriptions of the data elements based on the mappings. Constrained by our goal to replicate existing forms, we took the opposite approach where we start with ontological descriptions of the data elements, specify how they are used in assessment instruments as part of the description of instruments, and generate Web forms for the acquisition of data. Having the freedom to design their documentation system, Horridge *et al.* avoided the laborious work of manually modeling the domain concepts.

Semantic wikis extend regular wikis with semantic technologies, wherein each wiki article is an RDF resource, and an instance of some resource such as a class defined in the schema,[1] which can be asserted to have relations with other RDF resources. These relations are defined by the authors of wiki articles, which could be a challenging task to perform without previous knowledge of the domain or the modeling. In a survey of semantic wikis featuring OWL reasoning and SPARQL[2] querying facilities [4], a user

*To whom correspondence should be addressed: rafaelsg@stanford.edu

[1] The typical kinds of schema accepted are OWL and RDFS.

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

evaluation of a chosen semantic wiki implementation concluded that authoring instance data in such a way is cumbersome, even with users that were familiar with ontologies. A good solution to this would be exploiting the relations defined in the schema to provide "wiki article templates" whose form input fields derive from those relations, thus making it easier to author semantic wiki articles.

3 APPLICATION DOMAIN

Clinical functional assessment provides the application motivation for our work. Functional assessment is the evaluation of an individual's ability to perform body functions (e.g., flexing a joint) and defined tasks (e.g., walking a specific distance). It is necessary for evaluating disabilities for rehabilitation, for social security payment, or for decisions to retain or discharge service members who may be injured on duty. Despite its importance, it is not usually supported by electronic health record (EHR) systems [1]. These assessments are often documented using assessment instruments (e.g., check-lists and validated questionnaires) such as Karnofsky Performance Status [11]. Too frequently the data derived from using these instruments are saved as either blobs or non-standard data elements. While a standard such as LOINC® (Logical Observation Identifiers Names and Codes) defines the syntactic structures of assessment instruments as a hierarchy of panels with questions that have coded answers [10], it does not relate the semantic content of the questions and answers to standard terminologies and data models that allow meaningful querying and aggregation of acquired data.

In our application scenario we use, as exemplars, the U.S. Department of Veterans Affairs (VA) Disability Benefits Questionnaires (DBQs). DBQs are used to evaluate service members' disabilities and to determine the benefits for which they are eligible. We start off with these DBQs as our initial form specifications, and design an ontology-based method for Web form generation and structured data acquisition, subsequently exemplifying how one would go about exploiting such data for immediate or *post facto* analyses.

4 MODELING

In order to capture the semantic distinctions that are needed in functional assessment, we developed a Clinical Functional Assessment (CFA) ontology that models the concepts and relationships that occur in functional assessment instruments. We developed information models for such instruments and for data captured in the instruments. We will show how the CFA ontology and information models inform the generation of data-acquisition forms and how the resulting data can be queried and aggregated. Our goal was to develop a set of light-weight ontologies and models with minimal ontological commitments, and postponing alignment with possible upper-level ontologies to the future. Existing ontologies, such as the Information Artifact Ontology (IAO),[3] do not provide a modeling of forms and questions that we could reuse. Furthermore, what we need is an information model that states, for example, that the structure of a "question" includes a specific text, not an ontology that models parts of information artifacts as ontological entities (e.g., modeling the text of a question as an instance of "textual entity" class). Our ontologies reference the International Classification of Functioning, Disability and Health (ICF),[4] developed by the World Health Organization (WHO), and other reference terminologies such as SNOMED CT.[5]

Imports structure The modeling tasks of this project involve describing different domain areas, leading us to create separate ontology files that can be re-used independently. In our specific application we use the full import closure as depicted in Figure 1.

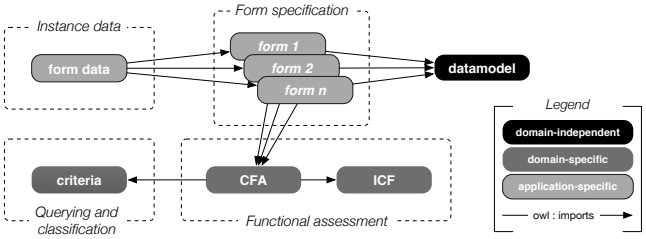

Fig. 1: Imports structure and role separation of ontologies developed for, or included as part of our modeling solution. Form specifications use terms from the *datamodel* ontology (e.g., to create question instances) as well as from domain-specific ontologies (e.g., CFA).

The ontology marked as *Instance data* in Figure 1 is the collection of data assertions from form submissions, possibly from different forms. The ontologies represented in *Form specification* are specifications of different forms; in our case, we use a single ontology that specifies two closely-related forms. The content of the above-mentioned ontologies is application-specific, that is, the way the data is represented is directly derived from the way in which forms are modeled (for different assessment instruments). However, resulting data still conform to the generic information models specified in the *datamodel* ontology. In this way, there is a separation of the *Form specification* ontologies (Abox axioms) from the *Functional assessment* ontologies that model the functional assessment domain and data models (mostly Tbox axioms). In *Querying and classification* we use a domain-specific ontology to apply SWRL rules,[6] and define complex OWL classes to facilitate querying in SPARQL and in OWL.

ICF ICF is a multi-purpose classification that, together with the International Classification of Diseases (ICD),[7] is a reference classification in the WHO Family of International Classifications (WHO–FIC). It provides a standard language and conceptual basis for the definition and measurement of functions and disability. However, unlike ICD codes that represent possible disease or injuries, coding different health and health-related states requires that ICF codes (e.g., "d4501" - walking long distance) be used in conjunction with component-specific qualifiers (e.g., a 0 to 4 scale to encode the range of impairment). Such a complex coding scheme makes it difficult to transform data derived from assessment instruments into the ICF format. Nevertheless, ICF provides a reference conceptual basis for the definition and measurement of functions and disability, thus justifying its usage in descriptions of functional assessment results, despite its limitations

[3] https://code.google.com/p/information-artifact-ontology

[4] http://www.who.int/classifications/icf/en

[5] http://www.ihtsdo.org/snomed-ct

[6] http://www.w3.org/Submission/SWRL

[7] http://www.who.int/classifications/icd/en

as a formal ontology [7]. To reference ICF concepts in our modeling of functional assessment descriptors, we use a version of ICF available from the National Center of Biomedical Ontology (NCBO) BioPortal repository [8], that is represented in OWL.

CFA The Clinical Functional Assessment (CFA) ontology models concepts and relationships that allow us to give formal descriptions of the findings, assessments, and measurements embodied in clinical functional assessment instruments. The ontology is divided into three main branches: (1) *Finding*: the result of an observation or judgement, (2) *Value* that defines collections of possible qualifiers and values for findings, and (3) *SubjectMatterOntology* that provides internally defined domain concepts that either are not available from standard terminologies or are references to standard terms that need to be organized into taxonomies. The *Finding* class is further subdivided into *Assessment* (those findings that have non-numeric result) and *Measurement* (those findings that have numeric results). We also define *FunctionalFinding* (a subclass of *Finding*) and *FunctionalAssessment* (a subclass of *Assessment*). In general, a functional assessment will have some assessed function that can be related to an ICF body function or activity (possibly as an exact match, specialization, or generalization), some assessed attribute, such as severity, that specifies the dimension of the function being assessed, and optionally some anatomical location of the assessment. Both findings and functions can be modified by qualifiers that further refine these entities. For example, a functional assessment may be made in the context of using assistive devices, and a function being assessed may have some temporal component (e.g., constant or intermittent pain). ICF being an imported ontology for CFA, all ICF categories, such as body structure, body function, activities and participation, and environmental factors are available for formalizing descriptions of functional assessments. For other standard terminologies such as SNOMED CT, ICD, and LOINC, instead of importing them as ontologies, we make references to them through an *ExternallyCodedValue* that specifies the terminology source and code. Queries that reference these codes require the availability of terminology services that relate these codes to other terms in the referenced terminologies.

The modeling of *Finding* is exemplified as follows, based on the "Back (Thoracolumbar Spine) Conditions" DBQ that we use as one of our exemplar assessment instruments; in the question on the severity of constant pain caused by radiculopathy on the right lower extremity, we define a subclass of *FunctionalAssessment* that has the assessed attribute 'severity', the assessed function 'icf:b2801 Pain in body part' that is qualified by a temporal quality 'Constant', and has anatomical location 'icf:s750. structure of lower extremity' with laterality 'Right'. Figure 2 illustrates the modeling of this assessment. With the modeling of the dimensions of assessment instrument questions, we can make queries on, and aggregate data collected through the instruments, as will be shown in Section 6.

```
cfa:hasAnatomicalLocation some
    ('icf:s750. Structure of lower extremity'
     and (cfa:hasLaterality value cfa:Right))
cfa:hasAssessedFunction some
    (cfa:Function
     and (cfa:isExactMatchOf some 'icf:b2801. Pain in body part')
     and (cfa:causedBy value cfa:Radiculopathy)
     and (cfa:hasQualifier value cfa:Constant))
cfa:SeverityOfPainSensationCausedByRadiculopathy
```

Fig. 2: Modeling of "severity of constant pain caused by radiculopathy in the lower right extremity".

Datamodel The *datamodel* ontology is a generic, context-free representation of a form (e.g., it models elements such as questions and sections) and the data generated from a form (e.g., a string value from a text area, or values from an enumerated value set). Figure 3 summarizes key aspects of our modeling: elements of a form are asserted as subclasses of *FormStructure*, such as *Form*, *Section* and *Question*. Each kind of *FormStructure* generates some kind of *Data*; every form submission generates an instance of *FormData*, which references (via the *hasComponent* property) all instances of *Data* generated in the process of parsing form answers. Specific sections such as *SubjectInfoSection* collect information pertaining to a subject, and these details are aggregated in an instance of *SubjectInformation*. An answer to an instance of *Question* gives rise to an instance of *Observation* with a *hasValue* property assertion to the IRI of the selected answer. An instance of *Observation* will be inferred to have an outgoing *hasFocus* property assertion if the *Question* instance it derives from encodes some kind of semantic description of the question's meaning via the *isAbout* relation. Each instance of *Question* specifies a set of possible (answer) values via a *hasPossibleValue* relation to a subclass of *Value*.

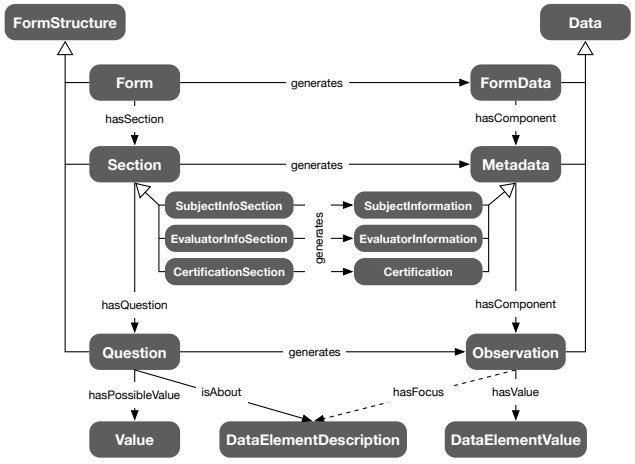

Fig. 3: Excerpt of the *datamodel* ontology classes and relations.

Form The *Form* ontology contains the set of individuals that are necessary to produce forms. While the technology we have developed is completely generic, we use as exemplars the U.S. Department of Veterans Affairs (VA) DBQs, which we modeled in an ontology named *DBQ*. This ontology contains instances of *Question*, *Section*, *Form* and other elements defined in the *datamodel* ontology (shown in Figure 3). Not only does this ontology rely on *datamodel* (for form structuring purposes), it also relies on functional assessment classes and individuals given in the *CFA* ontology, for example, values of a scale of severity of pain that should be presented as answer options to users reporting on the severity of constant pain in the lower extremity.

Criteria The *criteria* ontology contains SWRL rules to enrich the domain representation (e.g., if a *Question* instance has an *isAbout* relation with some instance i, then the *Observation* data instance that represents the answer to that question will get a *hasFocus* property filler i), as well as defined classes used to better support querying, which we describe in more detail in Section 6.

5 OWL-BASED DATA ACQUISITION

Our approach to data acquisition in OWL requires two components: firstly, an OWL representation (in the form of one or more ontologies) of the form structures (questions, sections, etc), and descriptions of those structures' meanings, and, secondly, the view component that is given by an XML file specifying user-interface aspects. So, in order to use our method, a user will have to model questions and their descriptions in OWL, and then specify the layout and content of the resulting form in XML.

We implemented our form generation and data acquisition tool in Java, using the OWL API v4.0.1,[8] and its source code is publicly available on GitHub.[9] The tool implementation and configuration details are omitted here due to lack of space, but can be found in the GitHub project wiki. The tool takes as input a user-defined XML configuration file, generates a form, and outputs form answers in CSV, RDF and OWL formats. The configuration file should contain a pointer to the ontology specifying the form, as well as its imports. The two major stages in the service are form generation and form input handling, as described below.

(1) Form generation – Steps to produce a form:
  (a) Process XML configuration, gathering form layout information, IRIs and bindings to ontology entities
  (b) Extract from the input ontology all relevant information pertaining to each form element:
    (b.1) Text to be displayed (e.g., section header, question text)
    (b.2) Options and their text, where applicable
    (b.3) The focus of each question
  (c) Generate the appropriate HTML and JavaScript code
(2) Form input handling – Once the form is filled in and submitted:
  (a) Process answer data and create appropriate individuals
  (b) Produce a partonomy of the individuals created in (2.a) that mirrors the layout structure given in the configuration
  (c) Return the (structured) answers to the user in a chosen format

The user-defined XML configuration (1.a) specifies: input and output information of the tool, bindings to ontology entities, and layout of form elements. The key XML elements are:

input: contains an *ontology* child element, and optionally a child element named *imports*
- ***ontology***: absolute path or URL to the form specification ontology (e.g., *DBQ ontology*)
- ***imports***: contains *ontology* child elements, which have an attribute *iri*, giving the IRI of the imported ontology

output: contains the following child elements
- ***file***: defines, via a *title* attribute, the title of the form. Optionally, a path can be specified within the *file* element where the HTML form file should be serialized
- ***cssStyle***: the CSS style class to be used in the output HTML

bindings: defines mappings to ontology entities, such as what data property is used to state the text of a question, or section headings

form: defines the layout and behaviors of the form

There is a wide range of versatility when configuring forms, such as: multiple levels of sub-questions, form element numbering, question type (e.g., radio, checkbox, dropdown, horizontal checkbox, etc), question-list layout (vertical or inline) and recurrence; one can specify that a collection of questions should be repeated any given number of times. Some more complex options include overriding the default (alphabetic) order of answer options, and triggering sub-questions when a specific answer is selected. These two features are exemplified in Figure 4: this question is configured with an attribute/value pair: *showSubquestionsForAnswer*="cfa:Yes" on the *question* XML element, so that answering 'Yes' triggers the sub-questions of that question. In Figure 4, under 'Right lower extremity', we have a question with a list of answer options derived from an enumerated value set, which would ordinarily be ordered alphabetically. However, 'None' would then appear between 'Moderate' and 'Severe', thus interrupting a severity scale. So we added: *optionOrder*="3;*" to the *question* element, which states that the would-be third option (alphabetically) should appear first, and the remaining (the "*" wild character stands for "all unmentioned options") should be presented in default order.

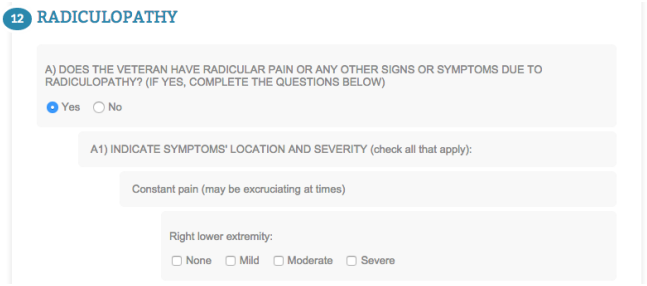

Fig. 4: The user interface of the form generated for the DBQ question corresponding to radiculopathy pain modeled in Figure 2.

The key output of the data acquisition tool is the OWL ontology, as it provides us with "semantically enriched" form data that can be used for aggregation and querying. The resulting data individuals are structured in OWL (via *hasComponent* relations) similarly to how the form is structured in the configuration, that is, if question Q is configured as having two sub-questions, then the *Observation* individual generated by Q will have two outgoing *hasComponent* relations to the instances of *Observation* generated by the two sub-questions of Q.

6 DATA ANALYSIS

One of the authors (Michael J. Tierney), who is a physician from the VA Palo Alto Healthcare System, validated the generated OWL-based versions of the DBQ forms, and filled in the "Back (Thoracolumbar Spine) Conditions" DBQ with 5 complete sets of sample data. The data gathered are stored in a graph database with support for SPARQL 1.1 querying and OWL 2 reasoning.

Since our data are both structured and semantically enriched, we are able to query the observations using SPARQL, classify them into criteria representing powerful OWL expressions, or manipulate them using SWRL. For example, Code Snippet 1 presents a simple SPARQL query that returns all instances of *Observation* where a patient presented signs or symptoms due to radiculopathy. It is worth observing that this query is formulated in such a way that it is independent of the assessment instrument, including the particular formulation of the question, but rather uses the appropriate focus individual from our CFA ontology.

[8] `http://owlapi.sourceforge.net`

[9] `http://github.com/protegeproject/facsimile`

Code Snippet 1 SPARQL query for retrieving all observations of radicular pain due to radiculopathy.

```
SELECT ?obs WHERE {
  ?obs a datamodel:Observation .
  ?obs datamodel:isDerivedFrom ?q .
  ?q a datamodel:Question .
  ?q cfa:isAbout
     cfa:signs_or_symptoms_due_to_radiculopathy .
  ?obs cfa:hasValue cfa:Yes }
```

In order to query for all observations of severe pain anywhere in the lower extremity, one could formulate an OWL DL query such as that given in Code Snippet 2.

Code Snippet 2 OWL DL query for retrieving all observations of severe pain anywhere in the lower extremity.

```
datamodel:Observation and
cfa:hasValue value cfa:severe and
cfa:hasFocus some (cfa:Assessment and
    (cfa:hasAssessedFunction some
        (cfa:isExactMatchOf some
            'icf:b2801. Pain in body part')) and
    (cfa:hasAnatomicalLocation some
        'icf:s750. Structure of lower extremity'))
```

In response to the query in Code Snippet 2, a DL reasoner uses the semantic descriptions of the observation foci, which are derived from the questions' *isAbout* property, to aggregate answers for severe pain for different parts of the lower extremity.

7 DISCUSSION

In this paper we presented a framework for OWL-based form generation and data acquisition that gathers form answers as tab-delimited data, RDF triples, or OWL instances, which can be subsequently analyzed in a systematic way (as shown in our queries in Section 6). Once the raw data is processed (by deriving the foci of observations from the *isAbout* field of the questions), the resulting data have no dependency on specific questions (except for provenance tracking), so if the form specification is modified, then previous form data are still comprehensible and sound (i.e., upon form specification changes the new data and old data remain compatible). However, if a user requires data to be structured in a different or more specialized format than ours, then either the software needs modifying, or a post-processing step would be necessary. The value of data in such a structured format in any arbitrary domain is twofold: automating, or improving the automation of the process of arriving at desirable conclusions from questions in the form, and for further analysis, for instance, via querying. In the clinical functional assessment domain, our modeling of forms and questions is consistent with the format of assessment instruments defined in LOINC. However, the types of queries we formulated for functional assessment data are unfeasible using LOINC, since LOINC provides no semantics behind what an answer to a specific question means.

We presented our modeling of functional assessments and assessment instruments, and demonstrated (1) how to generate forms and acquire data based on these OWL ontologies and data models, and (2) how to make use of the data using queries on individual subjects and queries that aggregate population data. The modeling contributions include (1) *CFA*: a clinical functional assessment domain ontology that allows defining questions being asked in an assessment instrument in terms of a rich ontology that integrates standard terminologies such as ICF and SNOMED CT, and which provides the means for making detailed or aggregate queries on acquired data, and (2) *datamodel*: an information model that allows the specification of generic assessment forms and the format of structured data acquired through the instruments.

We have designed our output model to support the acquisition of structured data through Web forms, and for the potential to integrate the data inside EHRs. It is straightforward to transform the data we capture as instances of *Observation*, *Certification*, *EvaluatorInformation*, and *SubjectInformation* into, for example, Health Level Seven (HL7) Reference Information Model (RIM) standard compliant data [5]. Finally, we have shown that the problem of structured data acquisition can be suitably tackled using OWL; our solution, though applied to the clinical functional assessment domain for the context of this paper, is entirely generic, and can easily be applied to an arbitrary domain.

ACKNOWLEDGMENTS

This work is supported in part by contract W81XWH-13-2-0010 from the U.S. Department of Defense, and grants GM086587 and GM103316 from the U.S. National Institutes of Health (NIH).

REFERENCES

[1] Buyl, R. and Nyssen, M. (2009). Structured electronic physiotherapy records. *Int. J. of Med. Inf.*, **78**(7), 473–481.

[2] Eriksson, H., Puerta, A. R., and Musen, M. A. (1994). Generation of knowledge-acquisition tools from domain ontologies. *Int. J. of Human-Computer Studies*, **41**, 425–453.

[3] Gennari, J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubzy, M., *et al.* (2003). The evolution of Protégé: an environment for knowledge-based systems development. *Int. J. of Human-Computer Studies*, **58**(1), 89–123.

[4] Gonçalves, R. S. (2009). *Semantic Wiki for Travel and Holidays using OWL*. Master's thesis, The University of Manchester.

[5] Health Level Seven (2015). HL7 Reference Information Model. `www.hl7.org/implement/standards/rim.cfm`.

[6] Horridge, M., Brandt, S., Parsia, B., and Rector, A. (2014). A domain specific ontology authoring environment for a clinical documentation system. In *Proc. of CBMS-14*.

[7] Kumar, A. and Smith, B. (2005). The ontology of processes and functions: A study of the international classification of functioning, disability and health. In *Proc. of the AIME Workshop on Biomedical Ontology Engineering*.

[8] Noy, N. F., Shah, N. H., Whetzel, P. L., Dai, B., Dorf, M., *et al.* (2009). BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, **37**, 170–173.

[9] Rector, A. (2013). Axioms & templates: Distinctions & transformations amongst ontologies, frames & information models. In *Proc. of K-CAP-13*.

[10] Vreeman, D. J., McDonald, C. J., and Huff, S. M. (2010). Representing patient assessments in LOINC®. In *Proc. of AMIA*.

[11] Yates, J. W., Chalmer, B., McKegney, F. P., *et al.* (1980). Evaluation of patients with advanced cancer using the Karnofsky performance status. *CANCER*, **45**(8), 2220–2224.