Utilizing NCBO Tools to Develop an Ontology:
A new tool that leverages the NCBO BioPortal REST Services

Stephen J. Granite[1] and Raimond L. Winslow[1]
[1]The Johns Hopkins University, Baltimore, MD, USA

Abstract

The CardioVascular Research Grid (CVRG) project uses emerging standards for describing diverse types of biomedical data. However, the medical community has no comprehensive ontology or data model in use at the current time to describe the most commonly collected biomedical time-series data type in modern health care, the electrocardiogram (ECG). To address this need, the CVRG team leads a Driving Biomedical Project for the National Center for Biomedical Ontologies (NCBO), using NCBO tools to develop an ECG ontology. To help in this process, the CVRG team developed a tool for searching NCBO managed ontologies to discover ECG-related concepts. The tool uses NCBO Representation State Transfer services. The tool appends discovered concepts to an OWL file containing the ontology. The user reviews the OWL file in Protégé and selects the concepts to be incorporated. This tool is available through the CVRG Home Page (http://www.cvrgrid.org).

Introduction

Recently, the National Heart Lung and Blood Institute (NHLBI) established the CardioVascular Research Grid (CVRG) project, to develop and deploy resources to represent, federate, share, and analyze multi-scale heart disease data[1]. However, there is no comprehensive ontology or data model for describing the electrocardiogram (ECG) in use at the current time.

Through a Driving Biomedical Project (DBP) for the National Center for Biomedical Ontologies (NCBO), the CVRG team is using NCBO tools to develop an ECG ontology[2,3]. To aid in the process of ontology development, NCBO investigators have developed a number of tools and resources (e.g., Protégé, NCBO BioPortal and BioPortal Representation State Transfer (REST) services)[4-7].

In order to discover ECG-related concepts already contained in NCBO-managed ontologies, we have developed a software application that uses the BioPortal REST services to obtain pre-existing concepts and their definitions from the BioPortal. The tool then repackages the concepts in Web Ontology Language (OWL) format, citing the original ontology as the source, and appends them to a local OWL file[8].

Design
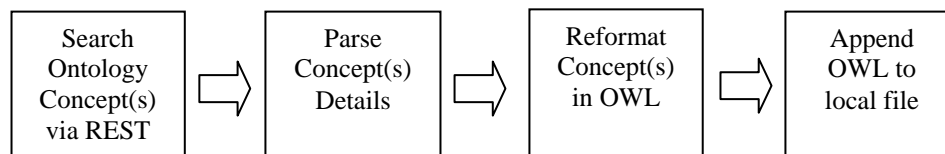
Figure 1. The Search & Append Tool Workflow.

The search & append tool workflow, shown in Fig. 1, is as follows:

1. Submit ECG-related search strings to NCBO BioPortal and return search results, which are matches that contain the search string, in an XML document.
2. Parse the XML document for pre-existing concepts, sources (identical concepts may be found in one or more ontologies), and definitions.
3. Re-format the parsed information into OWL.
4. Append OWL to a local file for use and further editing, using Protégé, by the ontology curator(s).

To derive this workflow, it is necessary to create components that work with NCBO BioPortal REST Services. These components must send queries to the services and accept XML responses. It is also necessary to create components that generate OWL representations of the XML responses, and that append those OWL representations to a local file.

Once the components exist, an interface must invoke the components in a sequential order. The interface must call the proper BioPortal REST service with a concept query. The interface must route the XML returned to the parsing mechanism. The interface must connect the parsed concepts with the OWL representation creator. The interface must route the OWL append function to the proper local file.

Implementation

The alpha implementation utilizes Java 1.5 as the development language, with Eclipse Europa as the integrated development environment[9,10]. The implementation uses REST services and Open Biomedical Services Ontology Access package provided by NCBO (Figure 2)[6,11].
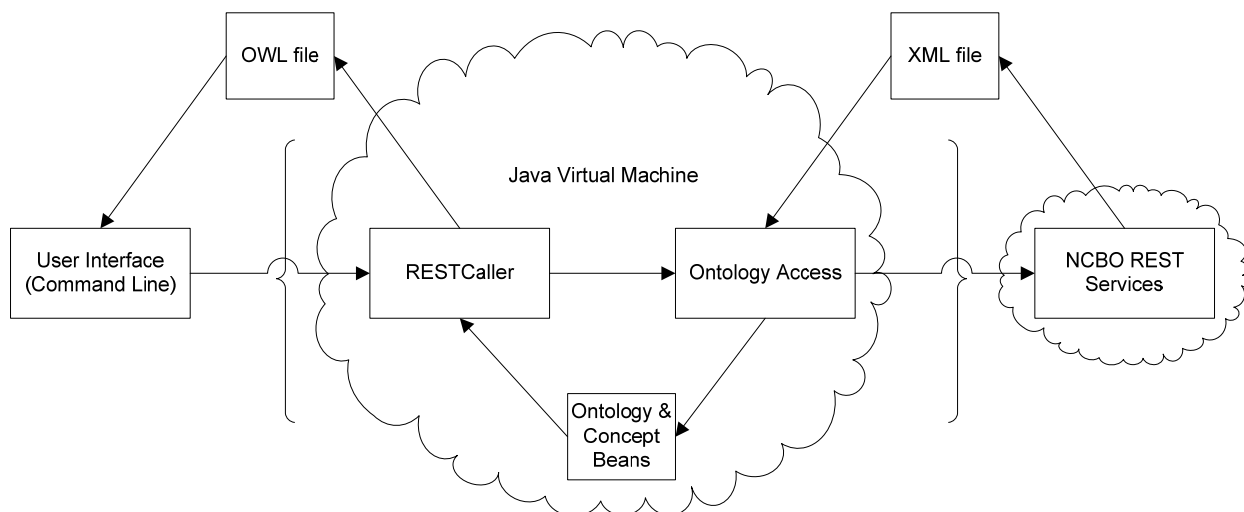
Figure 2. System Architecture & Data Flow.

The main program, invoked from Java, is RESTCaller. RESTCaller looks for five parameters provided in the command line invocation of the software. These five parameters specify: a) the name of the OWL output file; b) the directory in which the output will be stored; c) the ontology to be searched; d) the search string; and e) the name of the person performing the search. RESTCaller creates an output file in the location specified by the first and second parameters. With that file open, RESTCaller writes an OWL file header to prepare an OWL file output for the concepts retrieved. RESTCaller then calls the BioPortal ontologies REST service. RESTCaller parses the XML information returned by the BioPortal REST Service, and selects the information for the ontology named in the third parameter.
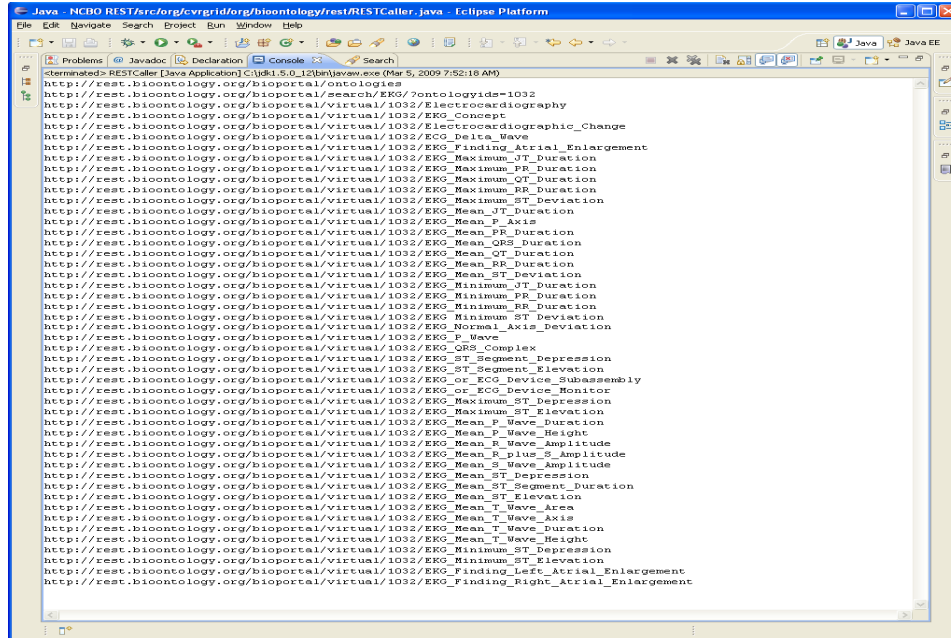
With that information, RESTCaller then calls upon the BioPortal search REST service for the search string in the fourth parameter for the ontology retrieved from the BioPortal ontologies REST service. RESTCaller then parses the XML information returned by the BioPortal search REST Service and collects all the concepts that contain the word. For each result returned, RESTCaller invokes the BioPortal concept REST service to retrieve all the details about each concept. RESTCaller then searches the details for concept definition entries. With the concept, its source (i.e., the ontology specified), and the definition, RESTCaller creates an object to store that information, along with the name of the file creator specified in the fifth parameter.

As RESTCaller retrieves and stores the respective concept information, it writes the same information in OWL format to the output file. Utilizing Dublin Core formatting, RESTCaller writes each concept with its id[12]. It also writes the file creator as dc:creator, the source as dc:source, and the definition as dc:description. To ensure that data integrity is not an issue, RESTCaller documents the time of concept information capture as dc:date. Once RESTCaller writes all the concepts in the output file, RESTCaller writes an OWL file footer, annotating the Dublin Core features used and ending the OWL information. RESTCaller then closes the file.

Once RESTCaller finishes writing the OWL output file, the ontology curator can use Protégé to edit the concepts returned in the output file.
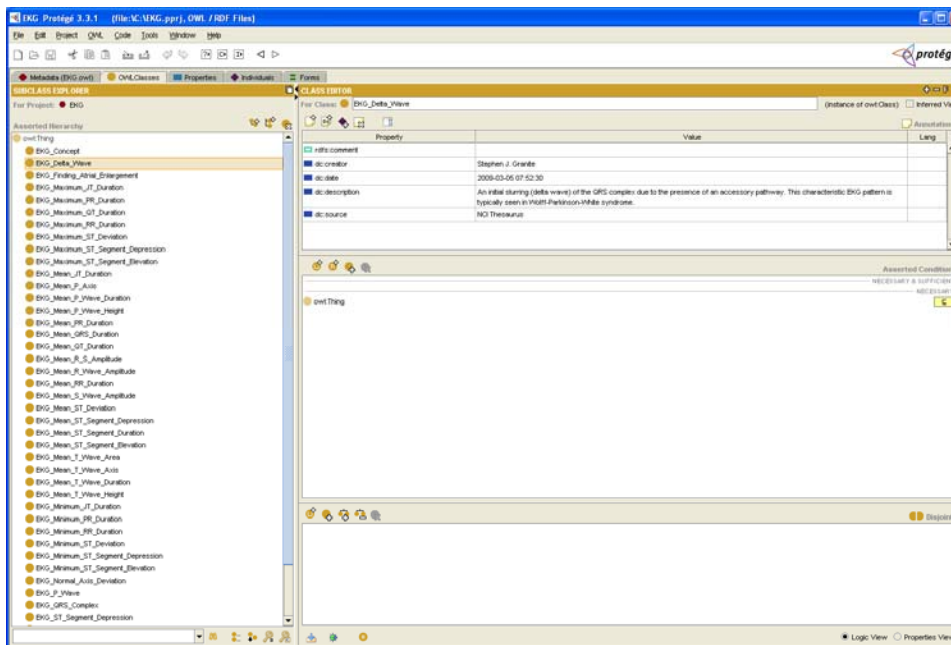
Use Case

In order to obtain pre-existing ECG concepts from the NCI Thesaurus, the tool should be executed with the proper parameters. In the case of a search for the concept "EKG", the parameters are a filename for the output (e.g., EKG.owl) and its location (e.g., C:\), the name of the thesaurus to search (i.e., NCI thesaurus), the concept to search (e.g., EKG) and the person producing the file (e.g., Stephen J. Granite). As shown in Figure 3, RESTCaller displays the REST service calls it makes in the command line console and produces the output file at the location specified.

Figure 3. REST calls made by RESTCaller for the concept "EKG" in the NCI Thesaurus.

Figure 4 shows the file produced follows proper OWL format viewed within Protégé. The item highlighted, EKG_Delta_Wave, shows the creator as the individual who created the file (e.g., Stephen J. Granite), the date of the information retrieval, the description of the concept and the source of the description.

Figure 4. OWL file produced by RESTCaller in Protégé, with EKG_Delta_Wave selected.

Conclusion

This paper describes the development of a tool for facilitating ontology generation as part of an NCBO DBP. Utilizing NCBO BioPortal REST services, the tool provides a mechanism to obtain pre-existing concepts and their definitions from the BioPortal. The tool then repackages the concepts in OWL format, citing the original ontology as the source, and appends them to a local OWL file. This allows one to obtain a subset of pre-existing ontologies quickly. This also enhances concept mapping to pre-existing ontologies because the concepts and their definitions are exactly the same.

The current interface in the alpha release of RESTCaller is command line based. To provide a better interface to RESTCaller, we will design a graphical user interface (GUI). A GUI will allow the user to enter the parameters and save them. The current interface only allows for searching one ontology per execution. A GUI could extend the number of ontologies searched, by allowing the user to select all, some or just one of the ontologies listed within the BioPortal.

The current interface uses application programming interfaces (APIs) presented by NCBO, but has not used the Protégé OWL API[13]. The API provides classes and methods to load and save OWL files, to query and manipulate OWL data models, and to perform reasoning. An implementation with the Protégé API would assist in preserving the hierarchical structure of the concepts returned from the NCBO REST services.

The alpha release of the RESTCaller is available via the CVRG website[2].

References

1. CVRG Home Page. Available from: http://www.cvrgrid.org/.
2.   National Center for Biomedical Ontologies. Available from: http://www.bioontology.org/.
3. CVRG NCBO Driving Biomedical Project. Available from: http://www.cvrgrid.org/?q=NCBO_Driving_Bio_Projects.
4. Protege. Available from: http://protege.stanford.edu/.
5.   National Center for Biomedical Ontologies BioPortal. Available from: http://bioportal.bioontology.org/.
6.   NCBO REST Services. Available from: http://bioontology.org/wiki/index.php/NCBO_REST_services.
7. Representational State Transfer. Available from: http://en.wikipedia.org/wiki/Representational_State_Transfer.
8. OWL Web Ontology Language. Available from: http://www.w3.org/TR/owl-features/.
9. Java Programming Language. Available from: http://www.java.com/en/.
10. Eclipse. Available from: http://www.eclipse.org/.
11. NCBO Open Biomedical Services (OBS). Available from: http://obs.bioontology.org/.
12. Dublin Core Metadata Element Set. Available from: http://en.wikipedia.org/wiki/Dublin_Core.
13. Protégé OWL API. Available from: http://protege.stanford.edu/plugins/owl/api/index.html.