Converting biological information to the W3C Resource Description Framework (RDF): Experience with Entrez Gene

Report

Lister Hill National Center for Biomedical Communications (NLM/ NIH)

By: Satya S. Sahoo

Email: sahoo@cs.uga.edu

Under the supervision of: Dr. Olivier Bodenreider

1. Abstract

The Entrez Gene (EG) database stores gene related data from sequenced genomes and of model organisms that are focus of active research [1]. We describe our experience in transforming the EG database into W3C Resource Description Framework (RDF) [2]. Our work is part of a larger effort to create a biomedical repository comprising not only information from structured resources (database and knowledge bases), but also from biomedical text (e.g., MEDLINE), of which information is extracted by SemRep [3]. Using the eXtensible Stylesheet Language Transformation (XSLT) [4] approach, we mapped the element tags of the EG XML representation to more intuitive relationship names manually, and used them during the automatic conversion to RDF. Finally, we store this RDF version of EG in the Oracle 10g [5] relational database with specific support for storing and querying of native RDF data.

2. Introduction

The NCBI databases store different aspects of biological information ranging from gene specific structured information (e.g., Entrez Gene) to textual data in biomedical publications (e.g., PubMed). The information retrieved from sources like Entrez Gene (EG) or the Online Mendelian Inheritance in Man (OMIM) [6] is represented in XML but follows different data type definitions (DTD). Hence, queries across the different National Center for Biotechnology Information (NCBI) data sources are only possible through implementation of complex software applications. Moreover, within one data source namely EG, a traditional relational database schema makes it extremely difficult to query for information using the relationships between the concepts. For example, EG does not include an element for ‘functional homology’, specifically in terms of coding for given proteins. Hence, it is extremely difficult to issue queries to search for functionally homologous genes (it will involve writing complex software that is very closely tied to the current DTD of EG, hence extremely sensitive to changes in the DTD).

The RDF format is a W3C recommendation to represent information in a machine understandable manner. An RDF repository consists of a set of assertions or triples. These triples are constituted of three entities namely, the *subject* – the triple pertains to this entity, the *object* – the entity that states something about the object and the *predicate* – the relationship between the *subject* and the *object*. The RDF format allows us to focus on the logical structure of the information in contrast to only representational format (XML) or storage format (relational database). Hence, we have implemented a workflow for conversion of EG information into RDF repository.

Our work is part of a larger effort to create a biomedical repository comprising not only information from structured resources (database and knowledge bases), but also from biomedical text (e.g., MEDLINE), of which information is extracted by SemRep [3]. This effort is also a contribution to the BioRDF task [7], an initiative of the Semantic Web Health Care and Life Sciences Interest Group at the W3C.

There are many issues involved in the conversion of XML data into RDF format including using unique identifiers, preserving of the original semantics of the data being converted, resolving bidirectional relationships and filtering redundant element tags from the original EG record. Unlike traditional XML to XML conversion, XML to RDF conversion should take into account the advantages of the RDF model in representing the logical structure of the information and the modeling of the relationships between concepts. The underlying objective of converting XML data into RDF is to capture the semantics of the data and leverage this semantic in querying the repository to not only retrieve the explicit but also the implicit knowledge through inference.

The rest of the report is organized as follows. Section 3 presents the implementation details; section 4 presents preliminary results. In section 5, we discuss the various issues involved in the transformation process and we conclude in section 6.

3. Implementation

We selected the use of XSLT based approach for converting the EG XML information into RDF as this allowed the separation of the application from the conversion logic. There have some efforts ("Ligand-Receptor Interaction, Molecular Interaction Networks, Ontology Evolution" sub-task in BioRDF project, http://esw.w3.org/topic/HCLSIG_BioRDF_Subgroup/Tasks)in converting biomedical data into RDF using XQuery [11] or XPath [9] language combined with XML parsers. Using the Java API for XML parsing (JAXP) as the platform to implement the application, we incorporated the XML to RDF transformation logic in the XSLT stylesheet. To store the RDF EG in the Oracle database, we used the Jena API [8] for conversion into n-triple format. Figure 1 illustrates the complete workflow implemented by us.

The XSLT stylesheet used in this workflow, using the XPath language, is specific to the EG database. As mentioned in section 1, we chose not to convert the element tags of the native EG XML representation mechanically into the *predicate* of the RDF triples. Instead, we manually converted the element tags of the native EG XML representation into meaningful relationship names that intuitively conveyed the semantics of the

connection between the *subject* and the *object*. For example, the element <*Gene-track_geneid*> is mapped to the more meaningful relationship named '*has_unique_geneid*'. This relationship also captures the uniqueness semantics of a '*geneid*' associated with gene record in EG. Moreover, there are multiple redundant elements tags as well as elements that formed multiple-layer containers around elements with actual attributes and values. We created a mapping between such element tags and the corresponding relationships, but we ignored the redundant or superfluous elements. For example, the element tag <*Date_std*> is repeated twice as part of the <*Gene-track_update-date*> and *Gene-track_create-date*> elements, which we ignore during the mapping process as they are redundant.

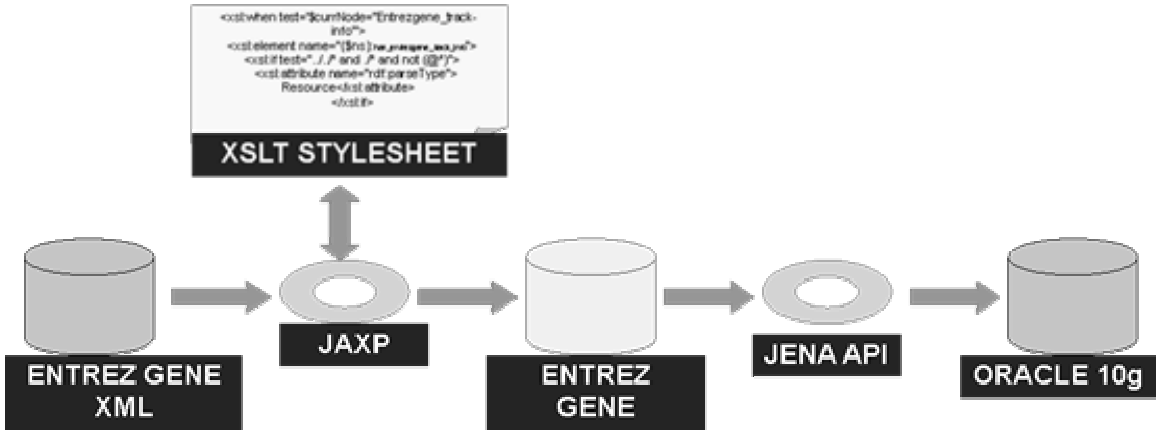

Figure 1: The implementation workflow for creating an RDF repository of the NCBI Entrez Gene database

We started organizing these relationships into an ontology, which currently reflects the native element tag nesting of the EG XML representation. Figure 2 illustrates the structure of the relationships using a simple *is-a* hierarchy. We aim to use this ontology as a repository of the knowledge used in creating the mapping between the element tags and relationships used as predicates in the RDF, not only for EG but also for other NCBI data sources. Additionally, we are exploring the inclusion of other relationships between these entities in addition to the *is-a* relationships. Oracle 10g is designed to store native RDF data and allows users to query these data using the native triple <*subject*, *predicate*, *object*> model. Using the Jena API, we converted the EG RDF store into *n-triple* format to populate the database. For example, the triple '*genid:ARP69702 http://www.ncbi.nlm.nih.gov/dtd/NCBI_Entrezgene.dtd/has_unique_geneid"351"*' to n-triple '*_:jA1<http://www.ncbi.nlm.nih.gov/dtd/NCBI_Entrezgene.dtd/has_unique_geneid>"351"*'.

The use of inference rules on this RDF data repository, using the native Oracle 10g interface, is the focus of our future work. We envision using this rule base to answer specific queries relating to the biomedical domain and leverage the information from all the NCBI data sources.

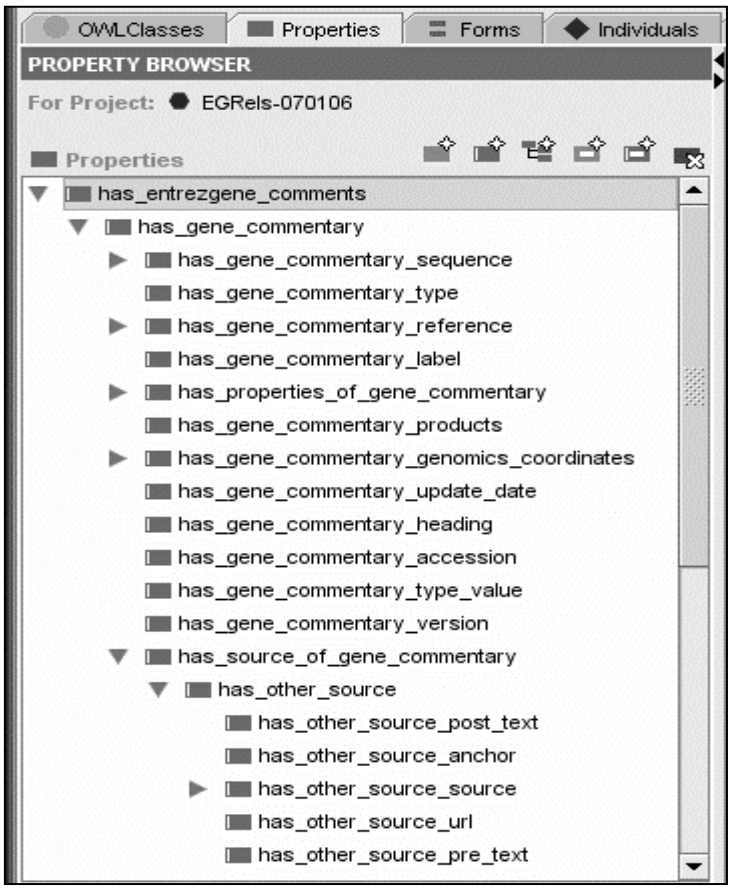

Figure 2: The hierarchy of relationships used as predicates in the RDF repository of Entrez Gene database

4. Results

We were able to successfully create an EG RDF information repository using the EG XML as the source of information. We created 106 intuitive relationships corresponding to the element tags of the native EG XML format. There were a total of 124 unique element tags. This involved considering elements with attributes as two distinct elements, one the element itself and another incorporating the attribute as part of the element name. For example <*Gene-track_status value*> and <*Gene-track_status*>, where '*value*' is the attribute of the element <*Gene-track_status*> as two distinct elements. We discarded repeated and superfluous element tags. Please see annexure A for a complete listing of the element tags and the corresponding relationships. We use the Java language based JAXP APIs to develop the application to implement the transformation and modeled the conversion logic in the XSLT stylesheet. Thus, our implementation model allows us to transform all the NCBI data sources into RDF by using data source specific stylesheets without changing the application code.

Initially, we used one EG record to prototype our approach. We converted the EG record for gene with EG 'gene_id' 351 to RDF. To ensure syntactic accuracy of the RDF file, we used the W3C web-based RDF validating application (http://www.w3.org/RDF/Validator/). The EG 'gene_id' 351 RDF gene record has 9245 triples.

For the complete EG data source, we started with 50 GB file in XML format (conforming to the EG DTD), which was converted to a 39 GB RDF file. The primary reason for this reduction in the size of the file is the ignoring of many elements in the original EG XML format. As mentioned in section 3 and this section, we used 106 elements out of 124 unique element tags in an EG record. Subsequently, we converted it to 33GB n-triple format file which was used to populate the Oracle 10g database. Till date, we have 411 million triples in the database. The population of the database is under progress at the time of writing this report.

5. Discussion

The primary objective of the RDF data model of the NCBI data sources, in this case the EG, is to faithfully model the logical structure of the data as present in the real world. It is not necessary that the specific structure specified in the DTD of EG accurately depicts that either in the nesting of the element tags or even the list of tags used to describe the gene data. The RDF model assumes a flat structure with respect to the *subject* being described and the different characteristics describing it, in turn. Hence, it is important to decide whether to reflect the native nesting of elements in the EG XML format or modify the structure to reflect one of the many possible perspectives of EG data. In our case, we chose to reflect the nesting of the element tags in the original EG data source. But, we believe that this may not necessarily be the best solution. For example, one possible approach may involve completely ignoring the nesting of the native EG XML format and listing all characteristics of a gene at the equal level for given gene.

The use of a specific identifier allows the unique identification of the nodes (*subject* and *object*) and *predicates* in an RDF repository. But, there is no globally accepted biomedical identifier schema that may be used. The bioinformatics community is currently debating this issue and there are many candidate schemas that may be used including the Life Science Identifier (LSID) [10] and solutions based on the HTTP protocol (i.e., URIs (Universal Resource Identifiers), URLs (Universal Resource Locators) and URNs (Universal Resource Names)). NLM resources such as the Unified Medical Language System could provide the basis for the identification of biomedical entities. As a temporary measure, we used the EG DTD URL (*http://www.ncbi.nlm.nih.gov/dtd/NCBI_Entrezgene.dtd/*) as the namespace to create the identifier of the entities in the RDF store. But, this may be changed with minimal effort, by modifying the XSLT stylesheet; thereby taking advantage of the inherent modularity of XSLT based transformations.

The relationships for corresponding element tags in the native EG XML representation were created manually. We anticipate further interactions in the community to evolve towards a more meaningful syntax of the relationships which may help in easier query formulation and execution. This will also take into account the capabilities and limitation of the Oracle 10g query interface, including the Java API based interface currently under development.

6. Conclusion

We demonstrate an implementation to convert NCBI EG data into a RDF repository using an XSLT approach. The transformation workflow decouples the application logic from the transformation logic by using the JAXP APIs for application development and the XSLT stylesheet for modeling the transformation logic. Further, instead of directly converting the element tags in the native EG XML representation, we manually map them to 106 intuitive relationships that we use as predicates in creating the RDF triples for EG data source. We use the Oracle 10g relational database that supports storage and retrieval of RDF data in its native format.

7. Acknowledgement

We acknowledge the contribution of Kelly Zeng in setting up the Oracle 10g relational database and populating it through conversion of the EG RDF store into n-triples. We also acknowledge May Cheh, Thomas C. Rindflesch, and Rob Logan for coordinating the summer program for graduate students at the Lister Hill Center (NLM/NIH).

8. Reference

1. Maglott, D., Ostell, J., Pruitt, K.D., and Tatusova, T., "Entrez Gene: gene-centered information at NCBI", *Nucleic Acids Res.* 2005 January 1; 33(Database Issue): D54–D58.
2. Resource Description Framework (RDF), http://www.w3.org/TR/2004/REC-rdf-primer-20040210/
3. Rindflesch, TC, Fiszman, M., "The Interaction of Domain Knowledge and Linguistic Structure in Natural Language Processing: Interpreting Hypernymic Propositions in Biomedical Text", *Journal of Biomedical Informatics*. 2003;36(6):462-77.
4. XML Schema Language Transformation (XSLT), http://www.w3.org/TR/xslt
5. Alexander, N., Ravada S., "RDF Object Type and Reification in Oracle"—Technical White Paper (http://download-east.oracle.com/otndocs/tech/semantic_web/pdf/rdf_reification.pdf)
6. Online Mendelian Inheritance in Man, OMIM (TM). McKusick-Nathans Institute for Genetic Medicine, Johns Hopkins University (Baltimore, MD) and National Center for Biotechnology Information, National Library of Medicine (Bethesda, MD), {date of download}. World Wide Web URL: http://www.ncbi.nlm.nih.gov/omim/
7. BioRDF subgroup: http://esw.w3.org/topic/HCLSIG_BioRDF_Subgroup
8. McBride, B. 2002. Jena: A Semantic Web Toolkit. *IEEE Internet Computing* 6, 6 (Nov. 2002), 55-59. DOI= http://dx.doi.org/10.1109/MIC.2002.1067737
9. XPath: http://www.w3.org/TR/xpath
10. Life Sciences Identifier (LSID) project: http://lsid.sourceforge.net/
11. XQuery: http://www.w3.org/TR/xquery/

Annex A

Entrez Gene Element Tag	*Relationship*
<Entrezgene_track-info>	has_entrezgene_track_info
<Gene-track>	has_gene-track
<Gene-track_geneid>	has_unique_geneid
<Gene-track_status value>	<------------------------------>
<Gene-track_status>	has_gene_track_status
<Gene-track_create-date>	has_creation_date
<Date>	<------------------------------>
<Date_std>	<------------------------------>
<Date-std_year>	has_year
<Date-std_month>	has_month
<Date-std_day>	has_day
<Gene-track_update-date>	has_update_date
<Date-std_hour>	has_hour
<Date-std_minute>	has_minute
<Date-std_second>	has_second
<Entrezgene_type value>	<------------------------------>
<Entrezgene_type>	has_entrezgene_type
<Entrezgene_source>	has_source_of_gene
<BioSource>	has_biosource
<BioSource_genome value>	<------------------------------>
<BioSource_genome>	has_genomic_source_of_gene
<BioSource_origin value>	<------------------------------>
<BioSource_origin>	has_biosource_origin_of_gene
<BioSource_org>	has_biosource_organism
<Org-ref>	has_reference_organism
<Org-ref_taxname>	has_taxonomy_name
<Org-ref_common>	has_common_name
<Org-ref_db>	has_organism_reference_database
<Dbtag>	<-------------------------------->
<Dbtag_db>	has_database_name
<Dbtag_tag>	<-------------------------------->
<Object-id>	<-------------------------------->
<Object-id_id>	has_object_id_value
<Org-ref_syn>	has_synonym_for_referred_organism
<Org-ref_syn_E>	has_synonym_for_referred_organism_E
<Org-ref_orgname>	has_organism_reference_name
<OrgName>	has_name_of_organism
<OrgName_name>	<-------------------------------->
<OrgName_name_binomial>	<-------------------------------->
<BinomialOrgName>	has_binomial_organism_name
<BinomialOrgName_genus>	has_genus
<BinomialOrgName_species>	has_species
<OrgName_lineage>	has_lineage_of_organism_name
<OrgName_gcode>	has_organism_name_gene_code
<OrgName_mgcode>	has_organism_name_mg_code
<OrgName_div>	has_organism_name_div
<BioSource_subtype>	has_biosource_subtype
<SubSource>	has_sub_source
<SubSource_subtype value>	<------------------------------>
<SubSource_subtype>	has_subsource_subtype

<SubSource_name>	has_subsource_name
<Entrezgene_gene>	has_entrezgene_gene_detail
<Gene-ref>	has_gene_reference
<Gene-ref_locus>	has_locus_of_gene_reference
<Gene-ref_desc>	has_gene_reference_description
<Gene-ref_maploc>	has_gene_reference_maplocation
<Gene-ref_db>	has_gene_reference_database
<Gene-ref_syn>	has_gene_reference_synonym
<Gene-ref_syn_E>	has_gene_reference_synonym_E
<Entrezgene_prot>	has_entrezgene_protein
<Prot-ref>	has_protein_reference
<Prot-ref_name>	has_protein_reference_name
<Prot-ref_name_E>	has_protein_reference_name_E
<Entrezgene_summary>	has_entrezgene_summary
<Entrezgene_location>	has_entrezgene_location
<Maps>	has_maps
<Maps_display-str>	has_maps_display_string
<Maps_method>	<------------------------------->
<Maps_method_map-type>	has_maps_method_map_type
<Entrezgene_gene-source>	has_entrezgene_gene_source
<Gene-source>	has_gene_source
<Gene-source_src>	has_gene_source_first_string
<Gene-source_src-int>	has_gene_source_integer
<Gene-source_src-str2>	has_gene_source_second_string
<Entrezgene_locus>	has_entrezgene_locus
<Gene-commentary>	has_gene_commentary
<Gene-commentary_comment>	has_comment_on_gene_commentary
<Gene-commentary_type>	has_gene_commentary_type
<Gene-commentary_type value>	<------------------------------->
<Gene-commentary_heading>	has_gene_commentary_heading
<Gene-commentary_accession>	has_gene_commentary_accession
<Gene-commentary_version>	has_gene_commentary_version
<Gene-commentary_seqs>	has_gene_commentary_sequence
<Seq-loc>	<------------------------------->
<Seq-loc_int>	<------------------------------->
<Seq-interval>	has_sequence_interval
<Seq-interval_from>	has_sequence_interval_from
<Seq-interval_to>	has_sequence_interval_to
<Seq-interval_strand>	has_sequence_interval_strand
<Na-strand>	<has_na_strand>
<Na-strand value>	<------------------------------->
<Seq-interval_id>	has_sequence_interval_id
<Seq-id>	<------------------------------->
<Seq-id_gi>	has_sequence_id_gi
<Gene-commentary_products>	has_gene_commentary_products
<Gene-commentary_label>	has_gene_commentary_label
<Gene-commentary_genomic-coords>	has_gene_commentary_genomics_coordinates
<Seq-loc_mix>	has_sequence_location_mix
<Seq-loc_whole>	has_whole_sequence_location
<Entrezgene_properties>	has_entrezgene_properties
<Gene-commentary_source>	has_source_of_gene_commentary
<Gene-commentary_properties>	has_properties_of_gene_commentary
<Other-source>	has_other_source
<Other-source_anchor>	has_other_source_anchor

<Other-source_pre-text>	has_other_source_pre_text
<Other-source_url>	has_other_source_url
<Other-source_src>	has_source_of_other_source
<Other-source_post-text>	has_other_source_post_text
<Gene-commentary_refs>	has_gene_commentary_reference
<Pub>	has_publication
<Pub_pmid>	has_publication_pmid
<PubMedId>	has_publication_pubmedid
<Gene-commentary_text>	has_gene_commentary_text
<Gene-commentary_update-date>	has_gene_commentary_update_date
<Gene-commentary_create-date>	has_gene_commentary_creation_date
<Entrezgene_comments>	has_entrezgene_comments
<Entrezgene_homology>	has_entrezgene_homology
<Entrezgene_unique-keys>	has_entrezgene_unique_keys
<Entrezgene_xtra-index-terms>	has_entrezgene_extra_index_terms
<Entrezgene_xtra-index-terms_E>	has_entrezgene_extra_index_terms_E
<Entrezgene_xtra-properties>	has_entrezgene_extra_properties
<Xtra-Terms>	has_extra_terms
<Xtra-Terms_tag>	has_extra_terms_tag
<Xtra-Terms_value>	has_extra_terms_value

Annex B

```
<?xml version="1.0"?>
<!DOCTYPE Entrezgene-Set PUBLIC "-//NLM//DTD NCBI-Entrezgene, 21st January 2005//EN" "NCBI_Entrezgene.dtd">
<Entrezgene-Set>
<Entrezgene>
<Entrezgene_track-info>
    <Gene-track>
      <Gene-track_geneid>351</Gene-track_geneid>
    </Gene-track>
</Entrezgene_track-info>
<Entrezgene_prot>
    <Prot-ref>
      <Prot-ref_name>
        <Prot-ref_name_E>amyloid beta A4 protein</Prot-ref_name_E>
        <Prot-ref_name_E>protease nexin-II</Prot-ref_name_E>
        <Prot-ref_name_E>A4 amyloid protein</Prot-ref_name_E>
        <Prot-ref_name_E>amyloid-beta protein</Prot-ref_name_E>
        <Prot-ref_name_E>beta-amyloid peptide</Prot-ref_name_E>
        <Prot-ref_name_E>cerebral vascular amyloid peptide</Prot-ref_name_E>
        <Prot-ref_name_E>amyloid beta (A4) precursor protein (protease nexin-II, Alzheimer disease)</Prot-ref_name_E>
      </Prot-ref_name>
    </Prot-ref>
  </Entrezgene_prot>
  </Entrezgene>
</Entrezgene-Set>
```

The Entrez Gene XML representation of the proteins coded by Gene with geneid 351 (representative fragment of XML with extra element tags to be valid XML)

Annex C

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:eg="http://www.ncbi.nlm.nih.gov/dtd/NCBI_Entrezgene.dtd/">
<rdf:Description rdf:about="eg:Gene-track_geneid/351">
<eg:has_entrezgene_protein rdf:parseType="Resource">
    <eg:has_protein_reference rdf:parseType="Resource">
      <eg:has_protein_reference_name rdf:parseType="Resource">
        <eg:has_protein_reference_name_E>amyloid beta A4 protein</eg:has_protein_reference_name_E>
        <eg:has_protein_reference_name_E>protease nexin-II</eg:has_protein_reference_name_E>
        <eg:has_protein_reference_name_E>A4 amyloid protein</eg:has_protein_reference_name_E>
        <eg:has_protein_reference_name_E>amyloid-beta protein</eg:has_protein_reference_name_E>
        <eg:has_protein_reference_name_E>beta-amyloid peptide</eg:has_protein_reference_name_E>
        <eg:has_protein_reference_name_E>cerebral vascular amyloid peptide</eg:has_protein_reference_name_E>
        <eg:has_protein_reference_name_E>amyloid beta (A4) precursor protein (protease nexin-II, Alzheimer
disease)</eg:has_protein_reference_name_E>
      </eg:has_protein_reference_name>
    </eg:has_protein_reference>
  </eg:has_entrezgene_protein>
  </rdf:Description>
</rdf:RDF>
```

The Entrez Gene RDF representation of the proteins coded by Gene with *geneid* 351 (representative fragment of RDF with extra element tags to be valid XML).