

A System for Linked Data Creation

Hyosook Jung¹, Seongbin Park²

¹ Department of Computer Science Education, Korea University,
Anam-dong, Seongbuk-gu, Seoul, 136-701 Korea,
{est0718, hyperspace}@korea.ac.kr

Abstract: Linked Data supports the integration between data by connecting resources on related topics. Authoring Linked Data is difficult because it demands the use of exact vocabularies and rules for making RDF data and linking RDF dataset. In this paper, we present a system that helps non-expert users create RDF documents and Linked Data easily. Using our system, a user can easily generate RDF documents and add new links between RDF entities without the complete knowledge about RDF grammar and vocabularies. We show how the proposed system can be used to create educational resources that need to integrate related resources from external dataset.

Keywords: Linked Data, education

1. Introduction

Linked Data [11] has been identified as an approach to disseminate and integrate data. It also offers new opportunities for the organization, integration, and retrieval of research results and educational materials. According to the Linked Data principles [8], URIs [15] are used for naming things and when someone looks up a URI, useful information is provided using the standard. While Linked data has been viewed as a promising approach to support machine-accessible dataset in diverse areas, in general it is difficult for non-expert users to create Linked Data in the format required by the Semantic Web [14]. For example, a user needs to identify a resource with a URI and represent it in RDF format [16] and it requires the knowledge about accurate vocabularies and rules of RDF.

In this paper, we present a system that helps non-expert users make new RDF documents and connect different RDF data to generate Linked Data.

To create Linked Data, a user needs to start with a set of RDF data. There are two ways to create RDF data. First, the user can create RDF data directly. Second, the user selects a certain Web resource while browsing Web resources and our system automatically presents the information of the resource in a simple RDF format. Since it is not easy to search available RDF data if the user is not familiar with RDF model and SPARQL [23], our system searches contents from different sources on the Web and exports them in RDF format.

Our system also helps searching information since in general if users are not familiar with the search domain, it is difficult to decide how

to define a query and how to navigate in the Web [4]. Specifically, our system shows a concept map which presents the relationships between keywords associated with the query when a user enters a query. Even if users do not have prior knowledge about a target domain, they can find concepts related to the desired information, acquire new knowledge, learn about the domain and organize their knowledge while browsing.

Once the user collects RDF data to create Linked Data, our system simplifies the processes for creating new links between RDF data by enabling users to add `<owl:sameAs>` and `<rdfs:seeAlso>` property that are generically used to create links between RDF entities. The users just select two RDF entities and click a button for `<owl:sameAs>` or `<rdfs:seeAlso>`. So, users do not have to define RDF structure and type specific RDF vocabularies.

This paper is structured as follows. Section 2 describes related works and section 3 explains the architecture of our system. The applications of the system are described in section 4 and section 5 describes experimental results. Section 6 concludes the paper.

2. Related Works

In this section, we describe related works to our research.

Google Refine [2] is a tool for working with messy data, cleaning it up, transforming it from one format into another, extending it with web services, and linking it to databases. While Google Refine deals with the static data prepared in advance, our system does the dynamic data selected according to user's

cognitive process while browsing. Google Refine is a tool for navigating and transforming tabular data such as CSV, TSV, Excel, etc. Although it supports connecting the data with other databases in order to add more value into the original data, it applies the various operations on the original data. However, our system helps users immediately save what they need while browsing resources from different sources such as Wikipedia [17], DBpedia [18], Google [19], etc.

RDF Extension for Google Refine [3] is a graphical user interface for exporting data of Google Refine projects as interlinked RDF data. Data can be reconciled against any SPARQL endpoint or RDF dump. The reconciled data can then be exported as RDF based on a template graph. Our system automatically generates RDF documents that describe web resources selected by the users. RDF Extension for Google Refine requires a skeleton which describes the form of the intended RDF in order to export tabular data into RDF. However, it is not easy for novice users because they should set a base URI, add prefix, RDF type and property, create new properties and classes, define namespaces and customize Google Refine Expressing Language expression to define URI values. In our system, users do not have to define how to align the RDF schema and RDF data. They just save the generated RDF documents.

Link creation between different dataset is a burden for users because it needs to specify manually and requests background information [5]. Although manual interaction for creating links is a tedious process, human verification is important especially when creating learning contents. Our system allows users with minimal knowledge about RDF to create links simply. That is, the users select the desired RDF dataset and clicks a button which adds links with `<owl:sameAs>` and `<rdfs:seeAlso>`. While [5] automatically generates new back links between RDF entities by using HTTP Referer information, it finds possible links within only a subset of RDF entities. In addition, it converts prepared sources into RDF and does not allow users to create new links.

[6] searches related dataset by matching with strings in a linking file containing descriptions of external resources and creates new links to external resources with `<owl:sameAs>`. Our system creates links to external RDF dataset as

well as existing RDF dataset with `<owl:sameAs>` and `<rdfs:seeAlso>`. While [6] adds `<owl:sameAs>` links to resources, our system creates links specified between RDF dataset as using two generic RDF properties. In addition, it converts prepared sources into RDF and it does not allow users to create new links.

To link related documents as Linked Data, the data should be represented into machine-processable data such as RDF/XML format. However, formal knowledge representation is not easy and it is also burden to consistently name scheme, accurately use RDF format, and fluently deal with new concepts and rules [7]. To minimize syntactic error, our system automatically creates information about a web resource in RDF format. It creates links between RDF dataset without human entering structured information about their linking.

Compared to related works, the features of our system are as follows.

1. It provides guided navigation in the Web for searching resources from different sources.
2. It converts not prepared sources but searched Web resources on the spot into RDF.
3. It automatically represents a resource in a simple RDF without user's definition about RDF form.
4. It allows users to create links between RDF entities with `<rdfs:seeAlso>` and `<owl:sameAs>`
5. It enables users to search RDF documents on the Web and save them.

Figure 1 shows how our work is related to other research.

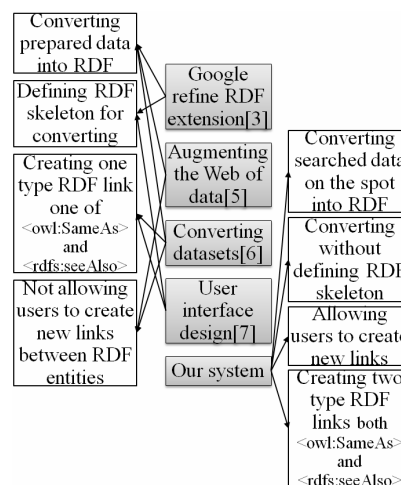


Figure 1. Related works to our research

3. Architecture

In this section, we explain the structure of the proposed system. We first describe the functionalities of the system. Then, we explain each module of the system.

3.1 Functionalities

There are four functionalities in our system.

1. Searching web resources

It provides a user interface that searches web resources from different data sources. Instead of showing a list of searching results, it shows related search keywords to the current entered query. The keywords are provided by the data sources such as DBpedia, Wikipedia, and Google. It allows users to navigate in search keywords related to the information that they wanted to find for. It displays the meaningful relation between the keywords like a concept map. When a user types a query, it shows relevant keywords by connecting them into the query like a map. If a user clicks 'search' button in one of the keywords, it shows new relevant keywords connected to the one. If the user clicks the original source of the keyword, it presents the web resources.

2. Generating RDF documents

It generates an RDF document that describes a web resource that is searched by keyword-based navigation system. It extracts the title and URL of a searched resource. Then, it transforms the title into the value of property `<rdfs:label>` and its URL into the value of property `<rdfs:isDefinedBy>`.

3. Creating links between RDF entities

It offers a user interface that allows users to create new links between RDF entities and finally generates linked data. It shows RDF entities defined in a RDF document and creates new links between the RDF entities that a user wants to connect each other. It enables the user to link two RDF entities by using `<rdfs:seeAlso>` and `<owl:sameAs>`. The user does not have to define new predicates. As linking the RDF entities in different RDF documents, the user can build linked data.

4. Searching RDF documents

It retrieves RDF documents by using Swoogle search engine [13]. When a user types a query, it shows a list of RDF documents searched

from Swoogle. It enables the users to save the documents in their local computers.

3.2 Modules

To implement functionalities described in section 3.1, we adapted functions of keyword-based navigation system [9] and those of linked data authoring system [10].

Figure 2 shows the structure of the proposed system.

The keyword-based navigation module starts with the searching result of user's query and continuously offers new keywords related to the query.

The Linked Data authoring module connects RDF documents that are either generated by the keyword-based navigation module or searched from external dataset to generate Linked Data. To create a link, the users just click a button which makes a new link between RDF documents without making full use of the RDF vocabularies and rules. The module create links by adding two predicates such as `<rdfs:seeAlso>` and `<owl:sameAs>`. `<rdfs:seeAlso>` is used to indicate a resource that might provide additional information about the subject resource. `<owl:sameAs>` indicates that two URI references actually refer to the same thing. It links an individual to an individual. Both of them can be used to interlink RDF data and help users find co-references between different data sets. Although the RDF files are expressed by different vocabularies, they can be integrated by our system.

The Concept map Generator presents a visualized diagram that shows relations between keywords. It also searches information from different sources such as Wikipedia, DBpedia, Google, etc.

The RDF Docs Generator represents a single resource that is found using the Concept map Generator in an RDF document. The RDF Docs Generator extracts the title and URL of the resource in order to use it as information about the resource. It assigns the resource a URI within the scope of the domain name of the Web server. It generates the RDF file that consists of RDF triples about the resource. That is, the resource is a subject, its title is an object of predicate `<rdfs:label>`, and its URL is an object of predicate `<rdfs:isDefinedBy>`.

Linked Data authoring module builds Linked Data by adding links between related RDF data.

RDF Docs Retriever requests Swoogle engine to search related RDF documents when a user enters a query and returns the result as a list of RDF documents. It enables the user to save the documents in a local computer. The user can use them in order to make Linked Data.

RDF Docs Parser analyzes the RDF document selected by a user into RDF triples [20] by using Protégé-OWL API [21] and shows a list of them as a table.

RDF Link Generator connects RDF data of one resource to that of another resource. It creates links between different RDF dataset by using `<rdfs:seeAlso>` and `<owl:sameAs>`. The former offers additional information about the subject resource. The latter identifies that two URI references actually refer to the same thing. Users choose RDF data from each RDF document and click a button `<rdfs:seeAlso>` or `<owl:sameAs>`. RDF Link Generator adds a new link between the RDF data and updates the RDF documents.

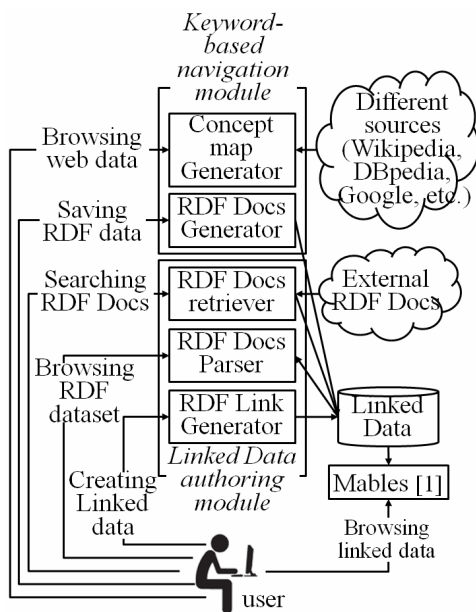


Figure 2. The process of the generation of Linked Data in our system

4. Application

In this section, we show how our system can be used in an educational environment. We also show that the proposed system can be easily combined with other systems that help users create and browse linked data.

4.1 Scenario 1

A teacher plans to teach the universe in a science class by using Web resources. The teacher wants to get students navigate Web sites related to the lesson and research the answers to the quiz. However, the teacher is concerned if the students get disoriented or cognitive overloaded while their browsing the Web sites because there are so many Web pages to explore. In addition, not all Web pages are associated with the lesson. The teacher wants to collect relevant Web pages, connect them and create learning content, but it is not simple to search available Web pages and organize them. So, the teacher decides to create Linked Data for the lesson by using our system.

First, the teacher finds Web pages by entering a query ‘universe’, ‘cosmos’, ‘star’, etc. (Figure 3)

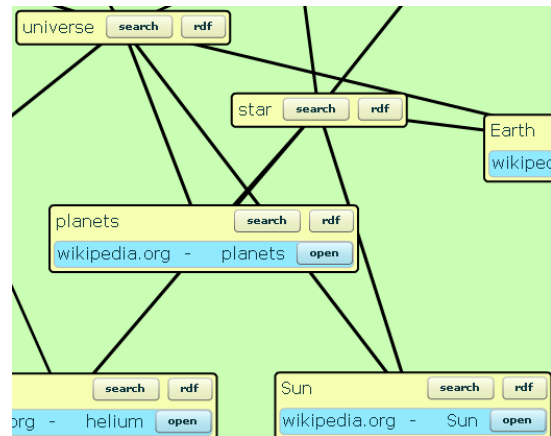


Figure 3. Starting entering queries in Keyword-based navigation system

The teacher continuously searches Web pages by clicking ‘search’ buttons inside the boxes that contain the keywords. (Figure 4)

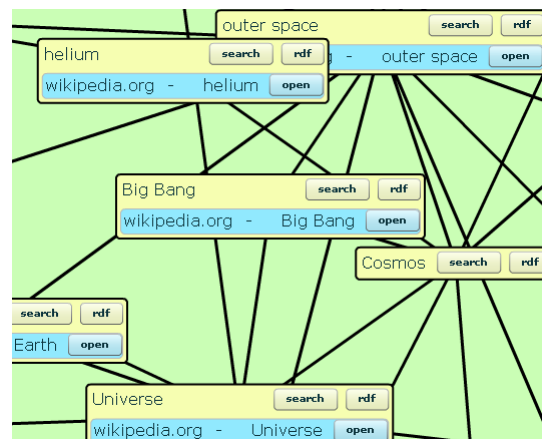


Figure 4. Selecting keywords in Keyword-based navigation system

The teacher can save the information about a certain Web page in RDF by clicking 'rdf' button located near 'search' button.

Figure 5 shows an RDF document about 'universe' Wikipedia page. It generates a unique URI for the page and saves its URL as a value of <rdfs:isDefinedBy> property and its title as a value of <rdfs:label> property .

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
xmlns:swrl="http://www.w3.org/2003/11/swrl#"
xmlns:protege="http://protege.stanford.edu/plugins/owl/prot
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:edu="http://163.152.111.73:8080/edu/linkedata.owl#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xml:base="http://163.152.111.73:8080/edu/linkedata.owl">
  <edu:Page
    rdf:about="http://163.152.111.73:8080/edu/Universe.rdf"
    <rdfs:isDefinedBy
      rdf:resource="http://en.wikipedia.org/wiki//Universe"/>
    <rdfs:label
      rdf:datatype="http://www.w3.org/2001/XMLSchema#stri
  </edu:Page>
</rdf:RDF>
```

Figure 5. RDF document about a Web page

After saving necessary RDF documents, the teacher creates new links between them by using Linked data authoring system. The teacher opens two RDF documents such as 'Star_system.rdf' and 'Galaxy.rdf'. The system shows a set of RDF triples defined in each document. The teacher selects RDF entities to be linked and clicks <rdfs:seeAlso>. The system inserts a new link connecting between two entities in the RDF document. As continuously creating links, the teacher can generate Linked Data. (Figure 6)

Figure 6. Creating a new link between RDF entities

Linked Data generated by our system can be navigated by Marbles [1] that displays all RDF triples describing a resource and follows <rdfs:seeAlso> and <owl:sameAs> found in retrieved data in order to obtain more information about a resource and human-friendly resource labels. For example, students access Marbles and enter the URI of an RDF document which is a starting point of Linked Data such as 'http://163.152.111.73:8080/edu/Star_system.rdf'. Marbles displays a view describing the RDF document. (Figure 7)

Figure 7. Displaying a view about an RDF

They can access the related Wikipedia page. (Figure 8)

Figure 8. Accessing Wikipedia page

They can also access a related RDF document which is connected by `<rdfs:seeAlso>` such as 'galaxy.rdf'. (Figure 9)



Figure 9. Accessing an RDF document linked by `<rdfs:seeAlso>`

Using our system, the teacher can get a simple RDF document that includes information about desired web resources. Unlike [3] and [7], the teacher does not have to manually define its RDF structure to describe the resource. In addition, the teacher can create links among RDF entities easily while [5] and [6] do not allow them to generate new links. The teacher also does not have to be worried about RDF syntax errors because simply clicking a button inserts `<rdfs:seeAlso>` and `<owl:sameAs>` that connect between related RDF entities.

4.2 Scenario 2

A teacher plans to teach students about earthquake and generates Linked Data about earthquake before the lesson. To this end, the teacher uses a system which helps users browse different information sources using OWL ontology [12] together with our system. (Figure 10)

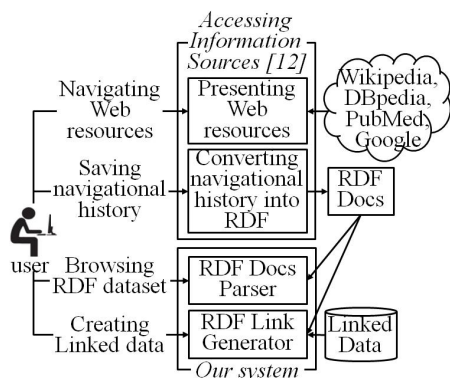


Figure 10. Combination of the propose system and [12]

While navigating in different web sites, the system records the navigational history that is a set of visited web pages. The teacher selects some interesting items in the history and saves them in RDF like earthquake.rdf. (Figure 11)

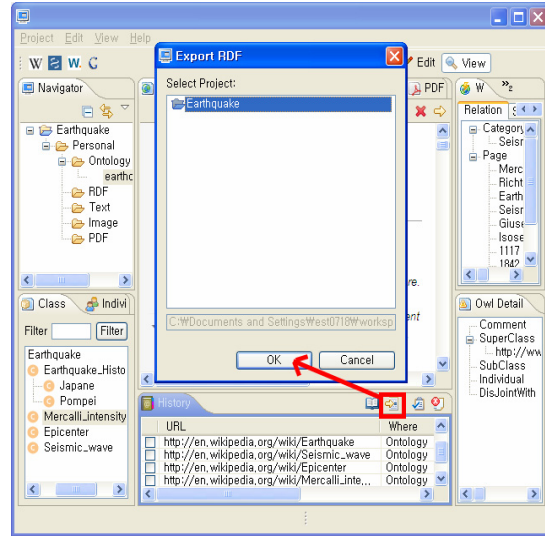


Figure 11. Saving the navigational history in RDF.

The teacher can open and edit the RDF file. (Figure 12)

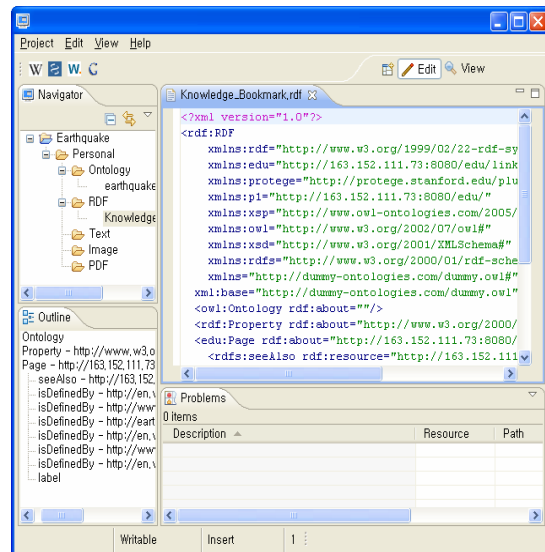


Figure 12. Editing RDF history file

The teacher navigates in web by using keyword-based navigation system and saves the information about searched web resources in RDF files. (Figure 13)

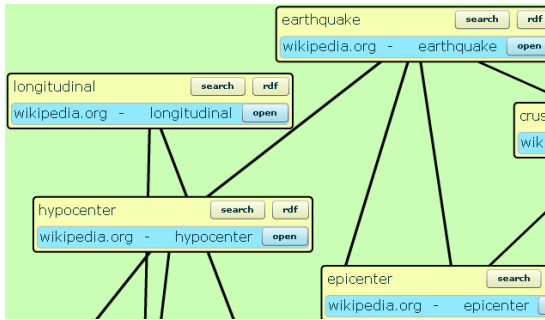


Figure 13. Searching information through keyword-based navigation system

The teacher saves information about a Wikipedia article about hypocenter like hypocenter.rdf. (Figure 14)

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
xmlns:swrl="http://www.w3.org/2003/11/swrl#"
xmlns:protege="http://protege.stanford.edu/plugins/owl/prot
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:edu="http://163.152.111.73:8080/edu/linkeddata.owl#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xml:base="http://163.152.111.73:8080/edu/linkeddata.owl">
- <edu:Page
rdf:about="http://163.152.111.73:8080/edu/hypocenter.rdf"
<rdfs:isDefinedBy
rdf:resource="http://en.wikipedia.org/wiki//Hypocenter" /
<rdfs:label
rdf:datatype="http://www.w3.org/2001/XMLSchema#stri
</edu:Page>
</rdf:RDF>
```

Figure 14. Saving RDF file of a web resource

The teacher runs Linked Data authoring system and opens two RDF files such as earthquake.rdf and hypocenter.rdf. The teacher creates a new link <rdfs:seeAlso> between two RDF entities such as Epicenter and Hypocenter. (Figure 15)

Figure 15. Creating a new link

The teacher lets students access Marbles and navigate in Linked Data. (Figure 16)

Figure 16. Navigating in Linked Data through Marbles

Linked Data created by our system can be visualized using RDF RDF Gravity [22] which offers a visualization of RDF graph structures and filter out. It also shows specific parts or fragments of RDF Graphs.

Figure 17 shows a screenshot that the linked data generated by our system is opened in RDF gravity. In the RDF graphs, we can see that 'earthquake.rdf' and is connected into 'hypocenter.rdf' through <rdfs:seeAlso>

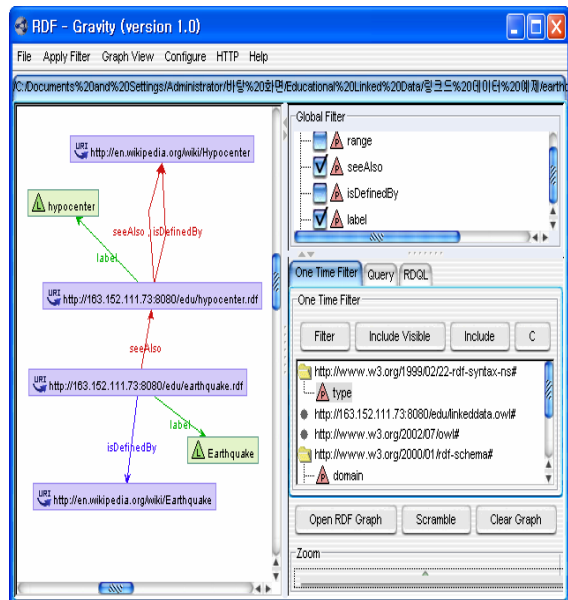


Figure 17a. Linked data in RDF Gravity

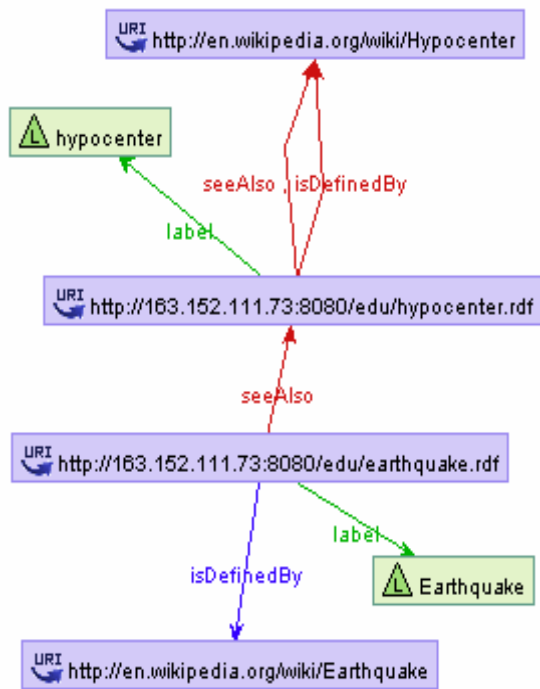


Figure 17b. Linked data in RDF Gravity

6. Experiment

We tested our system with several communities of graduate students majoring in computer science education. We made them choose one of topics such as semantic web, social networking service and Internet, and generate Linked Data about their topics using the system.

They commented on the system as follows;

<Advantage>

- It does not require the complete knowledge about RDF.
- It is scalable because it follows the standard for Linked Data such as using the unique properties of RDF for adding new links. So, it is easy to connect with other systems like Marbles.
- It helps users navigate in information without junk data because it provides resources related to a certain topic by using concepts.
- The users can create a simple linked data easily.
- Users can search RDF documents.
- It helps the users remember important concepts because it offers related keywords related to a certain domain.
- By using Marbles, the users can navigate RDF documents easily.

- Showing RDF triples makes it easy for the users to understand RDF entities and their relations.
- Users can link RDF entities easily by clicking buttons.
- Even though users cannot define a proper query, they can search resources because the system offers related concepts.
- Users can create an RDF document about a web resource although they are not experts about RDF or Linked Data.
- The non-expert users can easily create links between RDF entities because they just select proper entities and click a button.
- The users can efficiently browse resources and understand the content easily because Marbles presents the generated Linked Data simply.

<Disadvantage>

- It is not appropriate when they navigate in large web space.
- It makes limited relationships between RDF entities such as <rdfs:seeAlso> and <owl:sameAs>
- If the system offers a lot of concepts while searching web resources, it is too complex to understand the concept map.
- Users might not be familiar with RDF entities in Marbles comparing to HTML in web browsers. When it shows RDF entities, it needs to apply different style sheets to RDF document so that the users can understand it easily.
- Users should open two RDF documents, select RDF entities and add links between them. It might make the user tedious.

7. Conclusion

In this paper, we have presented a system by which non-expert users can create RDF documents and Linked Data easily.

We believe that if non-expert users contribute to production and consumption of Linked Data, it is possible to generate creative different Linked Data and can be helpful in expanding Linked Data.

The proposed system helps the users immediately to store web resources in RDF

format and easily to generate Linked Data by creating links between RDF entities. It supports guided navigation by a concept map. Users obtain new knowledge and learn about the domain as well as find the desired information.

Our system can be applied to generate learning content. Educators need to create proper learning content according to their lesson context such as students' background knowledge or interest, learning goal, time, place, etc. Most educators often utilize Web resources for their lesson. Even though there are a lot of resources on the Web, educators should find relevant resources and organize them before showing them to their students. If we help them search and connect resources, they might easily create abundant learning content. Our system enables the educators to generate Linked Data as learning content. By offering Linked Data which connects pieces of related data, educators can not only let their students explore customized learning content but also prevent their students from getting side-tracked.

REFERENCES

1. BECKER, C., C. BIZER, **Marbles**, 2009. <http://marbles.sourceforge.net>
2. Google, **google-refine**, 2011. <http://code.google.com/p/google-refine>
3. MAALI, F., R. CYGANIAK, **RDF Extension for Google Refine**, 2010. <http://lab.linkeddata.deri.ie/2010/grefine-rdf-extension>
4. MARCHIONINI, G., **Exploratory Search: From Finding to Understanding**, Communications of the ACM, Vol. 49, No. 4, 2006, pp. 41-46.
5. MÜHLEISEN, H., A. JENTZSCH, **Augmenting the Web of Data using Referers**, in Proceedings of the 4th Linked Data on the Web Workshop (LDOW 2011)
6. LEBO, T., G. T. WILLIAMS, **Converting Governmental Datasets into Linked Data**, in Proc. of the 6th International Conference on Semantic Systems (I-SEMANTICS '10), ACM, 2010, pp. 38:1-38:3.
7. DAVIES, S., J. HATFIELD, C. DONAHER, J. ZEITZ, **User Interface Design Considerations for Linked Data Authoring Environments**, in Proc. of the WWW2010 Workshop on Linked Data on the Web (LDOW 2010)
8. BERNERS-LEE, T., **Linked Data – Design Issues**, 2006. <http://www.w3.org/DesignIssues/LinkedData.html>
9. HWANG, C., H. KIM, H. JUNG, S. YOO, S. PARK, **Development of a System for Keyword-based Navigation on Multiple Sites**, Journal of Korea Association of Computer Education, Vol.14, No.1, 2011, pp. 123-135.
10. JUNG, H., H. KIM, S. PARK, **Development of a Linked Data Creation System for Ordinary People and Application**, Journal of Korea Association of Computer Education, Vol.14, No.2, 2011, pp. 47-59.
11. HEATH, T., C. BIZER, **Linked Data: Evolving the Web into a Global Data Space** (1st edition), Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool, 2011.
12. SUN, D., H. JUNG, C. HWANG, H. KIM, S. PARK, **Accessing Information Sources using Ontologies**, International Journal of Computers Communications & Control, ISSN 1841-9836, 6(2), 2011, pp. 349-366.
13. DING, L., T. FININ, A. JOSHI, R. PAN, R. S. COST, Y. PENG, P. REDDIVARI, V. C. DOSHI, J. SACHS, **Swoogle: A Semantic Web Search and Metadata Engine**, the 13th ACM Conference on Information and Knowledge Management, 2004.
14. BERNERS-LEE, T., J. HENDLER, O. LASSILA, **The Semantic Web**, Scientific American Special online Issue, 2001.
15. BERNERS-LEE, T., R. FIELDING, L. MASINTER, **Uniform Resource Identifier (URI): Generic Syntax**, 1998, <http://labs.apache.org/webarch/uri/rfc/rfc3986.html>
16. TAUBERER, J., **What Is RDF**, O'REILLY XML.com, 2006. <http://www.xml.com/pub/a/2001/01/24/rdf.html>
17. <http://wikipedia.org>
18. AUER, S., C. BIZER, G. KOBILAROV, J. LEHMANN, R. CYGANIAK, Z. IVES, **Dbpedia: A Nucleus for a Web of Open**

- Data**, the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference, LNCS 4825, 2007, 11-15.
19. <http://www.google.com>
 20. KLYNE, G., J. J. CARROLL, **RDF triples, Resource Description Framework (RDF): Concepts and Abstract Syntax**, W3C Recommendation, 2004, <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210>
 21. <http://protege.stanford.edu/plugins/owl/api>
 22. **RDF gravity**
<http://semweb.salzburgresearch.at/apps/rdf-gravity/index.html>
 23. <http://www.w3.org/TR/rdf-sparql-query/>