

# Finding Similarity and Comparability from Merged Hetero Data of the Semantic Web by Using Graph Pattern Matching

Hiroyuki Sato, Kyoji Iiduka, Takeya Mukaigaito, and Takahiko Murayama  
NTT Information Sharing Platform Laboratories, NTT Corporation  
3-9-11 Midoricho, Musashino-shi, Tokyo 180-8585 Japan  
{sato.hiroyuki, iiduka.kyo, mukaigaito.takeya, murayama.takahiko}@lab.ntt.co.jp

## ABSTRACT

We propose a method to find similarity and compared points from merged hetero data of the Semantic Web using graph pattern matching. A query pattern based on simple keyword is automatically created by analyzing the frequent occurrence of patterns of data structures and by using individual user context. Therefore, this query allows users to extract not only a subgraph that includes the keyword but also a cluster of characteristic data related to the keyword or user profile and preference. We call the proposed method context structure matching (CSM). We have been trying to apply CSM to a large amount of office data.

## 1. INTRODUCTION

With the advance of the Semantic Web, many people and organizations have been producing a large amount of data based on the Resource Description Framework (RDF). The idea of the Semantic Web is to provide frameworks that allow the sharing and reuse of data across various applications. Providing models and syntaxes for knowledge representation makes information on the Web more usable by machines and raises the quality and possibilities of processing by automatic tools. The goal of the Semantic Web is to turn the World Wide Web (Web) into a huge database of well-defined data that is easily reused by different machines that do not require knowledge of each other's functions.

RDF is a graph-based data model. Multiple data published on the Web in RDF format can be merged by placing the same data resources at a single node in the graph representation. Many query processors that are suitable for obtaining information from such RDF graphs by using graph pattern matching have been developed.

We propose a method to find similarity and compare points from merged hetero data of the Semantic Web using graph pattern matching. In this matching, we also use ontologies to find semantically similar graph patterns as the result of inference. This method can provide search engine users with added value in the results of simple keyword searches. We also introduce a way to use personal background information, which is called context, for the graph structure matching. We call this method Context Structure Matching (CSM).

## 2. EXTRACTING INFORMATION FROM MERGED HETERO DATA

### 2.1 RDF

The RDF specifies an interoperable model for describing the semantic attributes of information resources that are identified by uniform resource identifiers (URIs). The RDF has already been standardized in a W3C Recommendation. An RDF statement consists of three elements: a "resource", a "property", and a "value," as shown in Fig. 1. The respective elements are also called "subject", "predicate", and "object." An RDF statement consisting of these three elements is also called a "triple". RDF has an abstract syntax that reflects a simple graph-based data model [1].

### 2.2 Merging Data which have Hetero Graph pattern

Multiple items of data published on the Web in RDF format can be merged by placing same data resources at a single node in the graph representation. Connections might thus be made between three items of data (G1, G2, and G3) published on different Web sites, as shown in Fig. 2. G1 and G2 are data described by using vocabulary of the Friend-of-a-friend (FOAF) project [2]. G3 are described by vocabulary of the RDF Site Summary (RSS) [3].

The FOAF vocabulary is used to describe information such as personal profiles. The intention is to facilitate the formation of associations among people with similar interests, and backgrounds. The FOAF vocabulary covers items such as interest and nearby location as well as name and organization. A user is able to adopt unique IDs based on e-mail addresses and represent "people I know". The result of multiple FOAF definitions is a graph structure, which might be said to represent a social network.

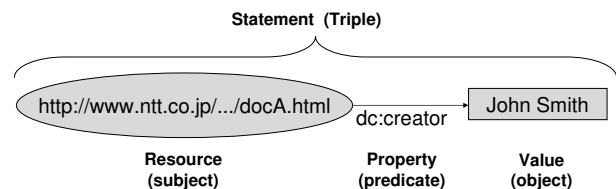


Figure 1: Graph representation of RDF data model.

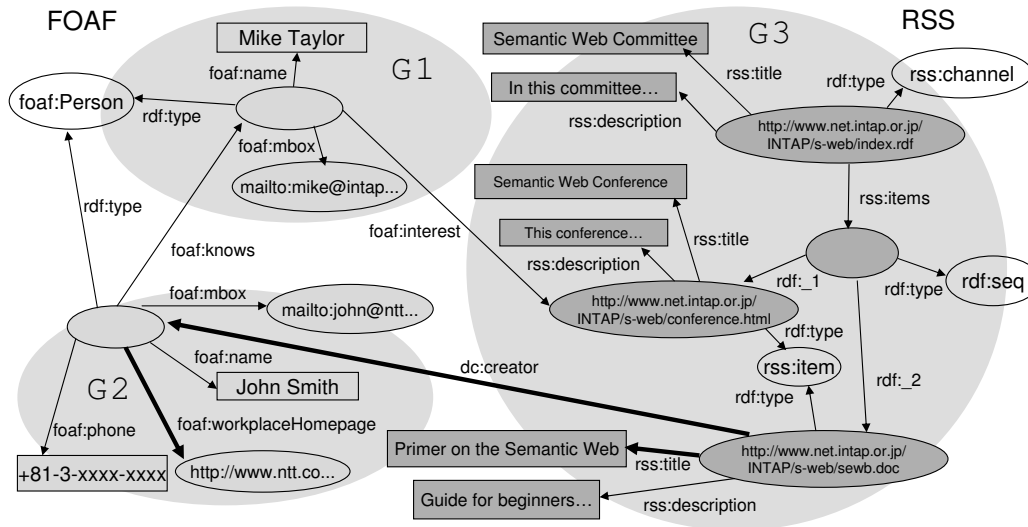


Figure 2: Merging of the RSS data and the FOAF data.

On the other hand, the RSS began as a channel-description framework/content-gathering mechanism, but is now often used for the delivery of news, personal comments, and/or personal diaries on shared Web sites. Users running an RSS client automatically obtain new information as it arrives or obtain information on a set schedule with the aid of RSS tags, which also let users rapidly publish commentary marked up with RSS information. Information from a single user is referred to as a “weblog”, and the weblog has become a major trend in personal Web publishing.

As mentioned above, the FOAF and RSS vocabularies have developed independently. Standard FOAF and RSS data have a hetero graph pattern, but both are represented by an RDF graph. They can be merged as a single graph, as shown in Fig 2. This would enable the computer making the connections to automatically answer queries such as “select a summary of Web pages that are of interest to a friend of ‘John Smith’ ” or “select the work place or phone number of the creator of ‘Primer on the Semantic Web’ ” by following arcs on the merged graph.

### 2.3 Query Processor and Query Language for RDF data model

Various tools for handling RDF have been developed. For example, RDF parsers [4] [5] [6], an RDF query engine, and database/API [7] make it easier for people to program Semantic Web applications. Tools such as Jena [8], Sesame [9], and Redland [10] incorporate all three of the above tool types.

An RDF parser syntactically analyses the statements in a given RDF file, even if the representation format of RDF is XML, N-Triple [11] or some other non-XML-based notation. An RDF database/API includes or is equipped with a facility for persistently storing a graph model derived from a parser. The query engine essentially gives the user a way to

obtain results for the matching of one graph pattern against the graphs stored in a database.

Various query languages are currently under development. Prud’hommeaux and Grosz give a list of about 20 existing query languages [12]. ICS-FORTH also provides an evaluation and comparison of languages and associated tools for the storage and querying of RDF-based data, with a focus on ontology querying [13]. To provide applications with uniform access to RDF data, the RDF Data Access Working Group (DAWG) within the W3C has been holding discussions on a standard query language for RDF named SPARQL since February of 2004 [14]. This will lead to the definition of an HTTP and/or SOAP-based protocol for selecting instances of subgraphs from an RDF graph.

The queries that we mentioned in subsection 2.2 can be described by using a query graph pattern. The example below shows a SPARQL query [15] to find the workplace of the creator of “Primer on the Semantic Web” from the information in the RDF graph of Fig. 2.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rss: <http://purl.org/rss/1.0/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?workPlace
WHERE (?document rss:title "Primer on the Semantic Web")
      (?document dc:creator ?person)
      (?person foaf:workplaceHomepage ?workPlace)
```

The query mainly consists of two parts, the SELECT clause and the WHERE clause. The SELECT clause identifies the variables of interest to the application, and the WHERE clause has triple patterns. This query contains a basic graph pattern of three triple patterns, each of which must match for the graph pattern to match. In this example, this query graph pattern and the graph pattern that is indicated by a bold line in Fig. 2 match, so the result of the query will be the value that corresponds to the variable workPlace.

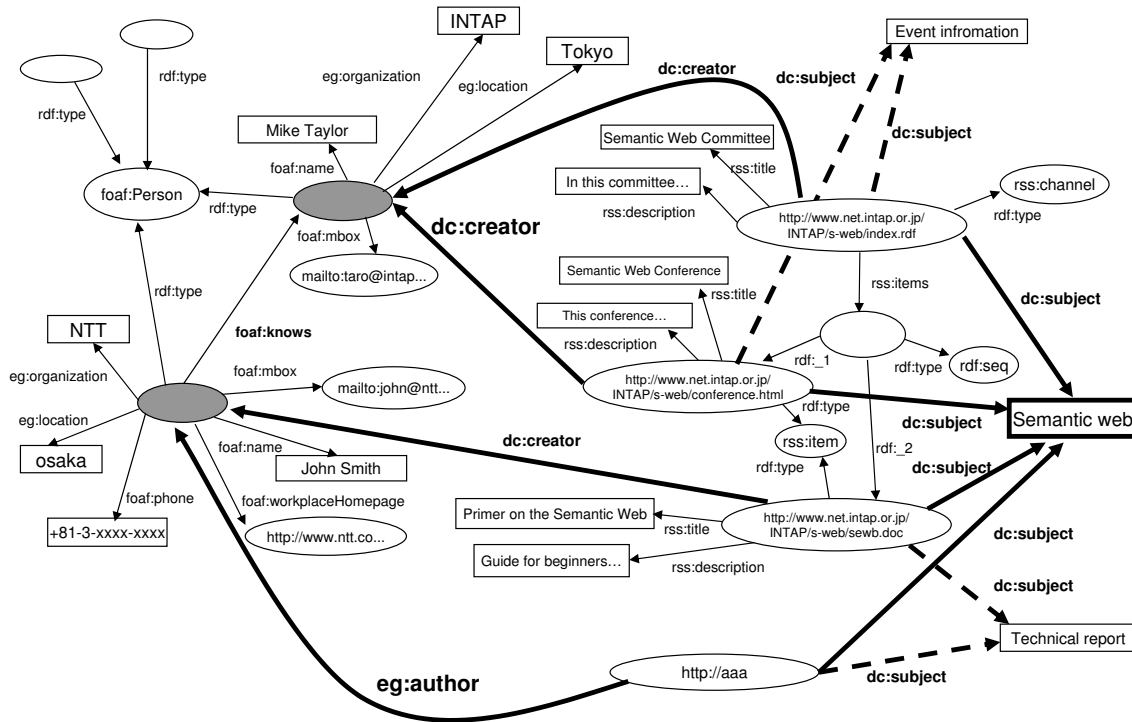


Figure 3: Graph queried by common query pattern.

### 3. PROBLEMS OF EXTRACTION

As mentioned in subsection 2.3, by using query graph pattern, users can extract information from RDF graphs composed of a large amount of triples. Users can make a complex query pattern, but they cannot make it if they do not know about the graph structure in advance. Considering that the service gives information from a database storing different hetero graph data patterns, not all end users understand all graph data structures of query targets. It is also difficult for most end users to modify a query pattern by themselves when new data is added to database.

Therefore, we propose a method to automatically create a new query graph pattern based on a simple keyword that users input. The query allows users to obtain information that includes a characteristic subgraph related to a keyword at the same time. This method allows users to provide not only a subgraph that includes the keyword but related information by extracting data from an RDF database using the following types of queries.

- Query that is made based on a frequent occurrence pattern related to nodes including keyword.
- Query that is enhanced from the above query by using the context of the individual user.

### 4. CREATING QUERY

This section describes how our method automatically creates queries and how we solve the above problems.

#### 4.1 Extracting Characteristic Frequent Occurrence Pattern

In this subsection we show an example of how our method extracts related information clusters from merged multiple graph structured data by using only a keyword that the user inputs.

An RDF graph of a query target, which is composed of the graph of Fig. 2 and other new data, is shown in Fig. 3. This example assumes that a user searches for information using the keywords “Semantic web” against the graph of Fig. 3. First, this method searches a node whose value includes the keyword. Next, paths between the node and instance nodes of important concept are searched. Important concepts should be predetermined by the user or service provider and are represented as a class of RDF Schema in RDF data. In this example, we define the class labeled “foaf:Person” as an important class. Instances of the class “foaf:Person,” which represent different individuals, are shown as gray circular nodes in Fig. 3. Bold lines in Fig. 3 indicate that there are two structured multiple paths between literal node “Semantic web” and each of the instance nodes that are the same. The same structure means that corresponding arc labels, which are properties of RDF, are also the same. This extracted pattern is called a common query pattern.

In this pattern matching, even if two corresponding properties are different, they may match in case there is an ontology that defines two different properties that are equivalent. In this example, we assume there is an ontology that defines the relationship between properties dc:creator and eg:author as



INTAP	eg:organization ←	(Mike Taylor)	dc creator ←	http://www.net.intap.or.jp/ INTAP/s-web/index.rdf	dc:subject →	Event information	Semantic web
				http://www.net.intap.or.jp/ INTAP/s-web/conference.html			
NTT	eg:organization ←	(John Smith)	dc creator ←	http://www.net.intap.or.jp/ INTAP/s-web/sewb.doc	dc:subject →	Technical report	
			eg:author ←	http://aaa			

Figure 6: Table representation of the result of query.

## 6. CONCLUSION

We explained a method to find similarities between people and compared points from merged hetero data of the Semantic Web using graph structure matching. We will continue to consider a method to extract data by using ontology and develop an application to efficiently handle a large quantity of RDF data.

## 7. REFERENCES

- [1] G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. <http://www.w3.org/TR/rdf-concepts/>
- [2] the friend of a friend (foaf) project. <http://www.foaf-project.org/>
- [3] G. BeGED-Dov, D. Brickley, R. Dornfest, I. Davis, L. Dodds, J. Eisenzopf, D. Galbraith, R. V. Guha, K. MacLeod, E. Miller, A. Swartz, E. van der Vlist. RDF Site Summary (RSS) 1.0. <http://web.resource.org/rss/1.0/spec>
- [4] J. Carroll. ARP: Another RDF Parser. <http://www.hpl.hp.com/personal/jjc/arp/>
- [5] D. Beckett. Raptor RDF Parser Toolkit. <http://www.redland.opensource.ac.uk/raptor/>
- [6] The Validating RDF Parser (VRP). <http://139.91.183.30:9090/RDF/VRP/index.html>
- [7] The RDF Schema Specific DataBase (RSSDB). <http://139.91.183.30:9090/RDF/RSSDB/index.html>
- [8] HP Labs, Jena 2 - A Semantic Web Framework. <http://www.hpl.hp.com/semweb/jena2.htm>
- [9] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *Proceedings of the International Semantic Web Conference 2002*.
- [10] D. Beckett. Redland RDF Application Framework. <http://www.redland.opensource.ac.uk/>
- [11] N-Triples W3C RDF Core WG Internal Working Draft. <http://www.w3.org/2001/sw/RDFCore/ntriples/>
- [12] E. Prud'hommeaux and B. Grosf. RDF Query Survey. <http://www.w3.org/2001/11/13-RDF-Query-Rules/>
- [13] A. Magkanaraki, G. Karvounarakis, T. Anh, V. Christophides, and D. Plexousakis. Ontology Storage and Querying. Technical Report No. 308, Foundation for Research and Technology Hellas, Institute of Computer Science, Information Systems Laboratory, April, 2002.
- [14] RDF Data Access Working Group. <http://www.w3.org/2001/sw/DataAccess/>.
- [15] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. <http://www.w3.org/TR/rdf-sparql-query/>