At present, a lot of information is serviced innumerably enough to arrive in saturation through internet. Nevertheless, searched information from web is not secured certainty. The biggest cause of these problem is that a man recognizes relations of terms by brain activity, but computer can not recognize and reason relations and meaning of terms[1]. Accordingly, current web is in the entering step of semantic web and various research in Ontology are progressed. Also, W3C adopted OWL (Ontology Web Language) for automatic construction into Ontology. Hence this study propose rules to convert RDB into Ontology description language OWL automatically.

1. Introduction

At present, a lot of information is serviced innumerably enough to arrive in saturation through internet. Nevertheless, searched information from web is not secured certainty. The biggest cause of these problem is that a man recognizes relations of terms by brain activity, but computer can not recognize and reason relations and meaning of terms[1]. Accordingly, current web is in the entering step of semantic web and various research in Ontology are progressed. Also, W3C adopted OWL (Ontology Web Language) for automatic construction of Ontology into recommendation in 2004.

OWL includes various vocabulary and formal semantics to express information in the semantic web, and supports more expression way, so it shows good performance to make out understandable information by computer[1].

OWL language provides three increasingly expressive sublanguages OWL Lite, OWL DL(Description Logic), OWL FULL[2,3].

OWL Lite supports those users primarily needing a classification hierarchy and simple constraint features.

OWL DL supports those users who want the maximum expressiveness without losing computational completeness and decidability of reasoning systems. OWL DL includes all OWL language constructs with restrictions such as type separation. OWL DL was designed to support the existing Description Logic business segment and has desirable computational properties for reasoning systems.

OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support every feature of OWL Full[2].

The rest of this paper is organized as follows: in the next section, we present related works to our research. The third section deals with the Conversion Rules. This section illustrates how to convert RDB into OWL. The final section summarizes the overall discussion and present future works.

2. Related Works

In recent years, there are three approaches have been reported. The first one is based in the semi-automatic generation of ontology from Relational DB model[4,9]. Then mappings are
defined between the DB and the generated ontology. The second approach, proposes the manual annotation of dynamic Web pages which publish DB content, with information about the underlying DB and how each content item in a page is extracted from the DB\[4,10\]. The third approach tries to map an existing DB to an appropriate existing ontology implemented in RDF(S) or OWL with the declarative language R2O\[4\]. But the first one does not allow the population of an existing ontology, which is a big limitation. The second approach does not deal neither with complex mapping situations and assumes we want to make our database schema public, which is not always the case although the third one R2O is automatically generating language, it does not solve the definition of degrees of similarity between DB elements and ontology very well and the readability of it can not be controlled easily by the human. Our approach is close to the first one.

As other research, T.S. Kim\[6\] proposed ontology generating algorithm, H.S. Kwak\[7\] proposed a conversion system HTML document into OWL ontology language, and S.H. Jang\[8\] proposed more detail approach than above.

Class hierarchies may be created by making one or more statements that a class is a subclass of another class. The Meaning of subclass in OWL is exactly the same: if the class description C1 is defined as a subclass of class description C2, then the set of individuals in the class extension of C1 should be a subset of the set of individuals in the class extension of C2\[2\].

3. Conversion Rules from RDB to OWL

In this section, we show a RDB schema(a company database) and relations of each tables, and present our converting rules.

3.1 Relation Database Schema

Figure 1 shows a database schema of relation database about company for our approach.

3.2 Relations of Tables

Via SQL, extract the table name from the RDB schema, and distinguish the entity table and relational table(see Figure 1).
3.3 Conversion Rules

Step 1) Distinguish and select RDB table from the existing DB schema.
Via SQL extract the entity table and relational table from the DB schema,
generating the entity table as Name Class with the same name of the corresponding 
table and generating the relational table as Property Class (see Figure 3).

Step 2) Generating subClass of the corresponding Class.

2.1) Via SQL extract the table which’s name is implicated in the other table, 
generating this table as the subClass of the corresponding Class.

2.2) When the foreign key referenced to the table of itself, generating the foreign key attribute as a Class which is a subClass of the Class of the referenced table.

Step 3) Generating the instances of primary key attribute as the Individuals of the corresponding class.

3.1) when the primary key of a table is composed within 2 attributes, then the foreign key attribute in them generates the ObjectProperty (see Step 4) the instances of other one generate the Individual of the table contains this kind of primary key.

Step 4) Generating the foreign key attribute as the ObjectProperty of the corresponding class.

4.1) when there are more than two primary key, primary key of other table is became ObjectProperty.

The domain of the ObjectProperty is the class of the table that contains this attribute, the range of it is the class of the referenced table.

Step 5) Generating the non-key attribute as the DatatypeProperty of the corresponding class.

The domain of the DatatypeProperty is the class of the table that contains the attribute, the range of it is one of the datatype of XML schema corresponding to the built-in datatype in the RDB schema.

Step 6) The relationship generating: (see Figure 4)

6.1) The 1:1 relationship generates the ObjectProperty with Cardinality=1.

6.2) The 1:n relationship generates the ObjectProperty with Cardinality=n.

Step 7) The n:m relationship generates the ObjectProperty. (see Figure 4)
when the table has the primary key composed within two attributes.

7.1) Each of the attributes is the primary of two different table, the domain of the property is the corresponding class of one table (e.g. EMPLOYEE), the range of the property is the other table (e.g. PROJECT) and vice versa.

7.2) Either of the attributes is the primary key of other table, the domain of the property is the corresponding class of the table (e.g. DEPARTMENT) which has this primary key attribute the range of the property is the class of the table (e.g. DEPT_LOCATIONS) that contains the two attribute and vice versa.

Step 8) Generating the subProperty of the corresp-
onding property
When the attribute name of a table is included in another attribute of other table, it generates the subProperty of the corresponding property.

In the Figure 1, underlined attribute means the primary key.

Table name EMPLOYEE, DEPARTMENT, DEPT_LOCATIONS, PROJECT, DEPENDENT are generated as each class name by Step1). Because table WORK_ON has the primary key composed within 2 attributes, and each of the attributes is the primary key of other table, so this table can not be a entity table, it is a relational table, it presents the relationship between two table which contain the attributes separately.

Step2) is a rule for when all of the table name are data of another table.

Attribute SSN in EMPLOYEE table, DNUMBER of DEPARTMENT, PNUMBER of PROJECT, PLOCATION of DEPT_LOCATIONS, DEPENDENT_NAME of DEPENDENT are generated as individual by step3).

DNUMBER of DEPT_LOCATIONS and ESSN of DEPENDENT are became ObjectProperty in Step3.1).

In the Step4), DNUMBER of EMPLOYEE is became ObjectProperty of EMPLOYEE class and MGRSSN of DEPARTMENT, DNUMBER of PROJECT, DNUMBER of DEPT_LOCATIONS and ESSN of DEPENDENT are became ObjectProperty for each class.

The other attributes, except primary key and foreign key, are became DatatypeProperty by Step5).

4. Conclusions and Future Works

We proposed several converting rules to generate ontology from RDB. From these rules, we could generate basic elements such as Class, subClass, individual, ObjectProperty, DatatypeProperty, and subProperty in OWL. But there are many other elements in OWL, so we need to extend our rules for generation of the other elements in OWL.

5. References


