Towards a grammar formalism for retrieving information on the Semantic Web

Eunhye Shin
Korea university
dc20005245@korea.ac.kr

Sujin Yoo
Korea university
mynameislydia@korea.ac.kr

Seongbin Park*
Korea university
hyperspace@korea.ac.kr

ABSTRACT

In this paper, we report our ongoing research on a grammar formalism for retrieving information on the Semantic Web. We view the Semantic Web as a collection of databases and use a two-level grammar by which one can specify context-sensitive constraints that need to be satisfied. To retrieve information, a user can simply specify keywords and our system can show a result that is a string derived using the two-level grammar.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query formulation; D.4 [**Information Systems Applications**]: Miscellaneous; H.5.4 [**Hypertext/Hypermedia**]: Architectures

1. INTRODUCTION

The Semantic Web is an environment where information is represented in a machine understandable way. One way to view the Semantic Web is that it consists of databases that can help computational agents perform various kinds of tasks [1].

In this paper, we propose an approach to write context-sensitive constraints using a grammar in order to retrieve information on the Semantic Web. To this end, we use a two-level grammar that is a 6-tuple (M, V, T, R_M, R_V, S), where M is a finite set of metanotions, V is a finite set of syntactic variables such that $M \cap V = \emptyset$, T is a finite subset of V^+, R_M is a finite set of metarules $X \to Y$, where $X \in M$, $Y \in (M \cup V)^{\square}$ or Y is a regular expression, and for all $W \in M$, (M, V, R_M, W) is a collection of context-free grammar and regular expression rules, R_V is a finite set of hyperrules of the form $H_0 \to H_1, H_2, \cdots, H_m$, where $m \geq 1$ and $H_0 \in (M \cup V)^+$, $H_i \in (M \cup V)^{\square}$ for $i \geq 1$, H_i is a hypernotion, and S is a string of positive length over $M \cup V$, respectively [2].

*Corresponding author

DIR 2013, April 26, 2013, Delft, The Netherlands.

The motivation of using a two-level grammar is that it allows to specify context-sensitive information in an intuitive way. Our approach allows a user to express facts that contain certain parameters which reflect structures of databases that constitute the Semantic Web. In other words, the formalism allows users to specify certain facts together with placeholders that can be instantiated using the data stored in databases.

The structure of this paper is as follows. Section 2 describes related research works. Section 3 explains the idea behind our approach using illustartive examples. Section 4 describes the structure of the system that we implemented. Section 5 concludes the paper and discusses research directions.

2. RELATED WORKS

There are two research areas that are related to our research. One is research about two-level grammar and the other is retrieving information on the Semantic Web. Two level grammar was introduced to define the syntax of ALGOL 68 by van Wijngaarden [3]. There are two types of rules in a two-level grammar. One is a metarule and the other is a hyperrule. A metarule is a context-free production rule and it can provide possible values for a metanotion in a hyperrule. A hyperrule can describe context-sensitive conditions and this is a machanism by which we can model constraints in a database or between databases. A two-level grammar has been used in specifying the syntax of a natural language which reflects grammatical constraints [4]. It has been also applied to define a programming language [2, 5]. One way to retrieve information on the Semantic Web is to use a query language, but traditional database query languages are not appropriate and users need a semantic query language such as SPARQL [6, 7]. In the mean time, for end users, a system such as SPARK [8] that can convert keyword queries into SPARQL queries can be helpful.

3. ILLUSTRATIVE EXAMPLES

In this section, we show how we can describe context-sensitive information using a two-level grammar. A string that can be derived using the grammar corresponds to certain information that can result from combining data that exist in the databases. There are two types of examples. The first example shows the case where we use one database that contains some number of tables. The second example shows how we can use two databases.

3.1 Example 1

Figure 1 shows the ER-Diagram of an example database (Travel database). There are five tables, where PK refers to a primary key and FK refers to a foreign key, respectively.

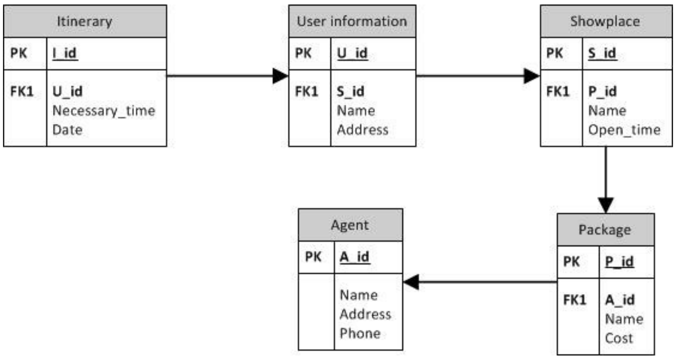

Figure 1: Travel database

Tables 1 to 5 show data stored in the database tables.

Table 1: Itinerary Table

I_id	U_id	Neccessary_time	date
350	1002	2 hours	2012-8-1
353	1001	5 hours	2012-7-2

Table 2: User information Table

U_id	S_id	Name	Address
1001	420	Sujin	Yangcheon-gu, Seoul
1002	401	Bob	Gangnam-gu, Seoul

Given this database, we can write metarules and hyperrules so that a string whose structure looks '( ), ( ) travels ( ) with ( ) package in ( ) agency.' can be derived as follows, where ( ) is a placeholder that can be instantiated with the data stored in the database tables.

The metarule is as follows.

```
I :: DATE, U_ID
U :: U_ID U_NAME travels S_ID
S :: S_ID S_NAME with P_ID
P :: P_ID P_NAME package in A_ID
A :: A_ID A_NAME agency.
```

The hyperrule is as follows.

```
START : I U S P A
where U_ID is U_ID : true
where S_ID is S_ID : true
where P_ID is P_ID : true
where A_ID is A_ID : true
```

Assuming that the tables contain data shown in table 1 to table 5, a possible derivation looks as follows.

$START \Rightarrow I\ U\ S\ P\ A \Rightarrow DATE,\ U\_ID\ U\ S\ P\ A$
$\Rightarrow 2012-7-2,\ 1001\ U\ S\ P\ A$
$\Rightarrow 2012-7-2,\ 1001\ U\_ID\ U\_NAME\ travels\ S\_ID\ S\ P\ A$

Table 3: Showplace Table

S_id	P_id	Name	Open_time
401	300	Buckingham Palace	11pm(10pm, Sun)
420	302	Hokkaido	11pm

Table 4: Package Table

P_id	A_id	Name	cost
300	204	15 days in West Europe	$2700
302	200	Hot Spring in Tokyo	$1400

$\Rightarrow 2012-7-2,\ 1001\ 1001\ Sujin\ travels\ S\_ID\ S\ P\ A$
$\Rightarrow 2012-7-2,\ Sujin\ travels\ S\_ID\ S\ P\ A$
$\Rightarrow \cdots \Rightarrow 2012-7-2,\ Sujin\ travels\ Hokkadio\ with\ Hot\ Spring\ in\ Tokyo\ package\ in\ Hana\ tour\ agency.$

Figure 2 shows how parameters in hyperrules can be instantiated and our system shows the string at the end of the derivation. The derivation starts by using the first hyperrule, START : I U S P A, where each of I U S P A is replaced by the corresponding metarule; i.e., I is replaced by DATE, U_ID, U is replaced by U_ID U_NAME travels S_ID, etc. A hyperrule which starts with "where" is applied in order to check context-sensitivity. For example, the U_ID from I (i.e., DATE, U_ID) and the U_ID from U (i.e., U_ID U_NAME travels S_ID) disappear when the hyperrule, "where U_ID is U_ID :true" is applied.

3.2 Example 2

In order to show how the same approach can be used with multiple databases, we added a university database that consists of two tables. In addition, we modified the travel database. Customers in the travel database are students in the university database. Buy table contains the information about package purchasing for each customer and user name of travel database corresponds to student name of university database (figure 3).

Tables 6 to 8 show the data contained in the database tables. Given these databases, we can write metarules and hyperrules so that a string whose structure looks '( ) is a Korea University student, majors in ( ) and takes a trip with ( ) package.' can be dervied as follows, where ( ) is a placeholder that can be instantiated with the data stored in the database tables.

The metarule is as follows.

```
RELATION :: USER STUDENT
UNI :: UNI.NAME DEPART
TRAVEL :: TRAVEL.NAME PACKAGE
```

Table 5: Agent Table

A_id	Name	Address	Phone_num
200	Hana tour	Seocho-gu, Seoul	02-993-2941
204	E agent	Bucheon, Gyeonggi	031-424-4421

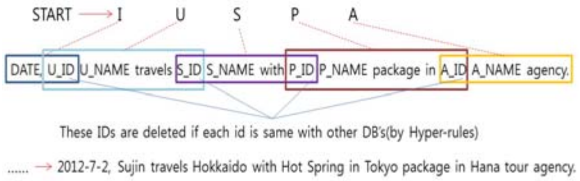

Figure 2: Derivation of a string

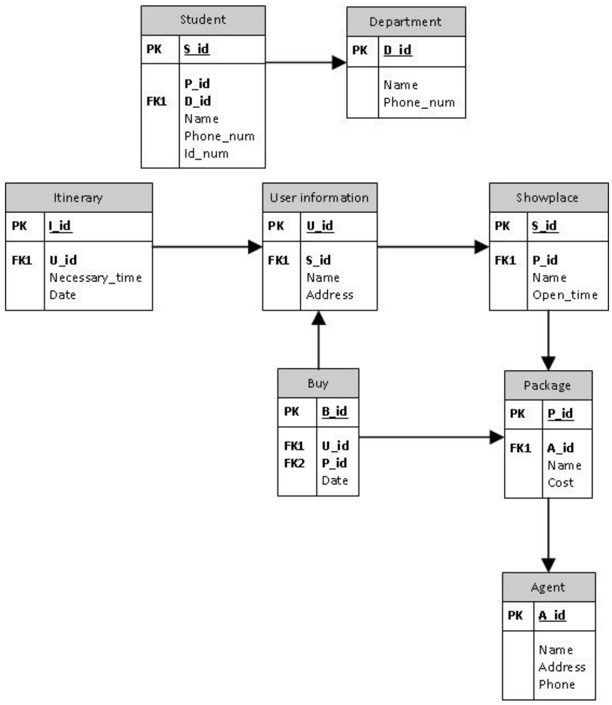

Figure 3: University database and modified Travel database

```
STUDENT :: S_Name
USER :: U_Name
UNI.NAME :: S_Name is a Korea University
            student, majors in D_ID
DEPART :: D_ID D_Name
TRAVEL.NAME :: U_NAME and takes a trip with
               U_ID BUY P_ID
BUY :: U_ID P_ID
PACKAGE :: P_ID P_NAME package.
```

There are five types of hyperrules.

(1) The following hyperrule is the start rule.

```
START : UNI RELATION TRAVEL
```

(2) The following hyperrules verify whether student name and user name are the same or not.

```
S_NAME of UNIV.NAME is same S_NAME of STUDENT
of RELATION, U_NAME of TRAVEL.NAME is same
U_NAME of USER of RELATION, S_NAME of STUDENT
is same U_NAME of USER, S_NAME of STUDENT,
U_NAME of TRAVEL.NAME and U_NAME of USER
: true
```

(3) The following hyperrules verify whether data is connected correctly in Univ DB.

```
D_ID of UNI.NAME is same D_ID of DEPART is
same, D_ID of UNI.NAME and D_ID of DEPART
: true
```

(4) The following hyperrules verify whether data is connected correctly in Travel DB.

```
P_ID of TRAVEL.NAME is same P_ID of PACKAGE,
P_ID of BUY is same P_ID of PACKAGE, P_ID of
TRAVEL.NAME, P_ID of PACKAGE and P_ID of BUY
: true
```

(5) The following hyperrule verifies whether purchasing data is connected correctly in Travel DB.

```
where U_ID is U_ID : true
```

Figure 4 shows how parameters in hyperrules can be instantiated and our system shows the string at the end of this figure.

Table 6: Buy Table

B_id	U_id	P_id	Date
465	1001	300	2011.11.01
274	1002	302	2011.10.10

Table 7: Student Table

S_id	P_id	D_id	Name	Phone_num	ID_num
1	150	100	Sujin Yoo	243-5678	051901
2	150	100	Bob	234-5784	011901

4. TLG SYSTEM

Our system (TLG sysetm) has been implemented using Java and HSQLDB system [9]. The operation starts by taking a keyword from a user. The Searching scale setting module assigns a column according to the keyword. The Result creating module matches the keyword against data in the assigned column. Finally, the Result printing module shows the result from result creating module as a string. Figure 5 shows the structure of the TLG system, where numbers inside small circles correspond to steps involved in sequence.

5. CONCLUSIONS

Table 8: Department Table

D_id	Name	Phone_num
100	Computer Science Education	32-1234
102	Mathematics	32-3456

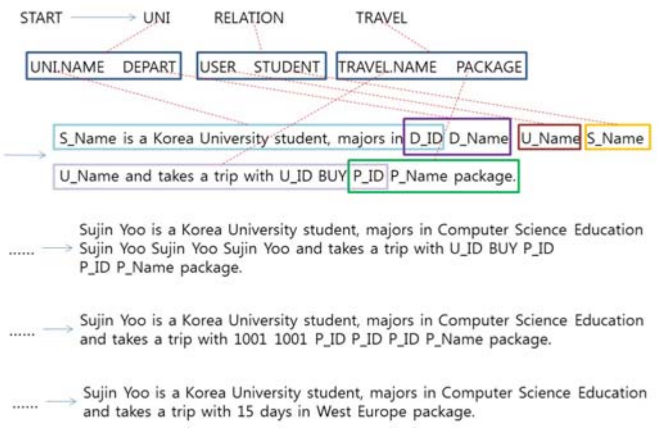

Figure 4: Derivation of a string

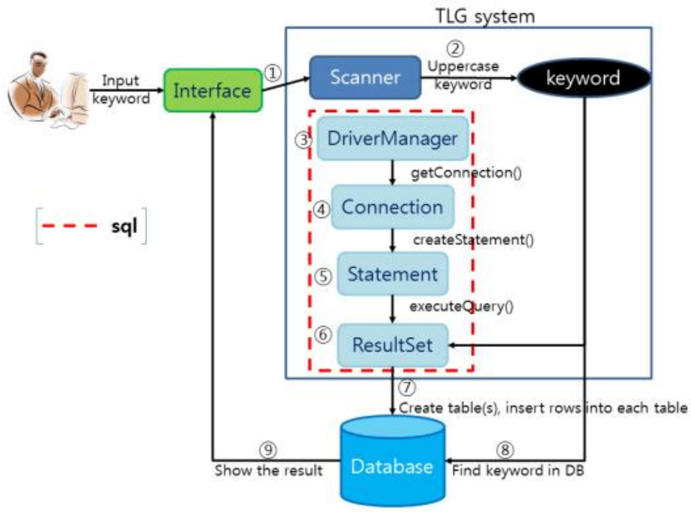

Figure 5: The structure of the system

In this paper, we report our ongoing research on how a two-level grammar can be used to retrieve information on the Semantic Web. We view the Semantic Web as a collection of databases and constraints existing in the colecction can be specified using hyperrules of the formalism that we proposed.

The motivation of the current research is that once we have a declarative description about the Semantic Web using a formal grammar, it becomes possible to *process* the Semantic Web. In other words, a part of the Semantic Web can be fed into a computer program as an input and the program can parse the input and perform some task. This is in line with the goal of utilizing the Semantic Web as a representation medium for computations [10].

Currently, we are implementing a system that parses expressions defined using a two-level grammar and shows derivation results. We are also working on ways by which the information on the Semantic Web can be exploited using a Semantic Web browser [11] and how to extend the idea of a Semantic Web expression [12] in the context of linked data [13].

6. REFERENCES

[1] C.C. Marshall and F.M. Shipman. Which Semantic Web?. In *Proceedings of the 14th ACM Conference on hypertext and hypermedia*, 2003.

[2] B. Edupuganty and B.R. Bryant. Two-level Grammar as a Functional Programming Language. In *The computer journal*, vol 32, no 1, 1989.

[3] A.van Wijngaarden. Report on the Algorithmic language ALGOL 68. Numer 14: 79-218, 1969.

[4] B.R. Bryant, D, Johnson, and B. Edupuganty. Formal specification of natural language syntax using two-level grammar. In *Proceedings of the 11th International Conference on Computational Lnguistics*, 1986.

[5] J. Maluszynski. Towards a Programming Language Based on the Notion of Two-Level Grammar In *Theoretical Computer Science*, 13-43, 1984.

[6] R. Fikes, P. Hayes, and I. Horrocks. OWL-QL-a language for deductive query answering on the Semantic Web In *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*. vol 2, issue 1, 19-29, 2004.

[7] M. Arenas, C. Gutierrez, D.P. Miranker, J. Pérez, and J.F. Sequeda. Querying Semantic Data on the Web. In *SIGMOD Record*, vol. 41, no. 4, 2012.

[8] Q. Zhou, C. Wang, M. Xiong, H. Wang, and Y .Yu. SPARK: adapting keyword query to semantic search In *Proceedings og the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference*, 694-707, 2007.

[9] HSQLDB *www.hsqldb.org*

[10] M.A. Rodriguez and J. Bollen. Modelling Computations in a Semantic Network. `http://arxiv.org/abs/0706.0022`, 2007

[11] Y. Kim, S, Yoo, and S. Park. A Semantic Web browser for novice users. In *Proceedings of the 6th International Conference on Complex, Intelligent, and Software Intensitve Systems*, 2012.

[12] E. Shin and S. Park. Two level grammar and the Semantic Web. In *Proceedings of the International Conference on Applied and Theoretical Information Systems Research*, 2012.

[13] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. In *International Jounral on Semantic Web and Information Systems*, 2009.