CDL-based Semantic Representation for Web Databases

Mamdouh Farouk, Mitsuru Ishizuka
Creative Informatics Department
Graduate School of Information Science & Technology, the University of Tokyo
Tokyo, Japan
mamdouh@mi.ci.i.u-tokyo.ac.jp, ishizuka@i.u-tokyo.ac.jp

***Abstract*— One important task toward making web machine understandable is representing web data into semantic formats. Since a huge amount of web data stored in databases, researchers pay much care to convert DB to semantic format. Most of the previous work depends on domain ontology to covert DB to RDF. However, depending on domain ontology has many problems. This paper proposes converting web DBs into CDL semantic format, which does not depend on domain ontology. In addition, CDL describes semantic structure of web content based on a set of predefined concepts and semantic relations. The first step to convert DB to CDL is representing DB schema into a semantic network. The second step is automatic generation for CDL based on the semantic network. A prototype is presented to show the effectiveness of the proposed approach.**

Semantic represenation, semantic database, Concept Description Language

I. INTRODUCTION

Semantic web is a vision for the next generation of the web in which web agents interact web resources like human. Moreover, representing web data and resources into a standard semantic format is an important step toward semantic web [1]. This semantic representation enables intelligent agents to interact web resources semantically [2][3][4]. Researchers pay much care about representing DB into semantic format because a huge amount of web data stored in Databases [2]. On the other hand, the process of converting DB to semantic format should be simple [5] to encourage the DB owner to convert his data.

There are different approaches to convert DB to a semantic format [2][6][7]. A common step in these approaches is finding a mapping between DB schema and ontology structure. Based on this mapping, the DB can be accessed semantically either by generating semantic representation corresponding to original data or by keeping the data in the DB, where it can be managed better, and generating semantic representation on demand. There are different approaches for the latter way. One approach is converting SQL query result to RDF on the fly when the DB is queried [3]. This approach is suitable in case of dynamic web pages that retrieve content from underlying DB. Another approach is developing a semantic access layer as an intermediate layer between web agents and normal DB [8].

Work that addresses the issue of representing web DBs into a semantic format uses semantic web languages (RDF, DAML, OWL ...), which depend on ontology [9]. However, this paper proposes a technique to represent web DBs content into a new semantic structure format, which does not depend on ontologies.

Many researchers use ontologies to represent web data into a machine understandable format [10][7]. However, there is no agreement on an ontology. In other words, there are many ontologies available on the Internet that may use same terminology to refer different concepts and vice versa [11]. Moreover, the problem of ontology interoperability is still an open problem. So using current web ontologies in representing web data is not an optimal solution.

This paper focuses on representing schema of an underlying DB of a website into a semantic format. This semantic representation contains detailed semantic relations of the database. Moreover, the proposed approach represents web database into CDL semantic format, which does not depend on ontologies. The generated CDL semantic representation together with semantic database schema can be used by intelligent search engines to answer queries and get accurate search results.

This paper is organized as follows: section 2 defines the semantic language, which used in the proposed technique. Section 3 explains the architecture of the proposed system. Section 4 shows a prototype for the proposed approach. Comparison with related work is discussed in Section 5. Finally, section 6 contains the conclusion of this research.

II. CONCEPT DESCRIPTION LANGUAGE (CDL)

Our approach uses a new semantic language, which called Concept Description Language (CDL). This semantic format, which proposed by Institute of Semantic Computing, describes semantic/conceptual structure of contents (resources) and can deal with natural languages, mathematical expressions, movie, music, etc [12]. The aims of CDL are to realize machine understandability of web text contents, and to overcome language barrier on the web [13].

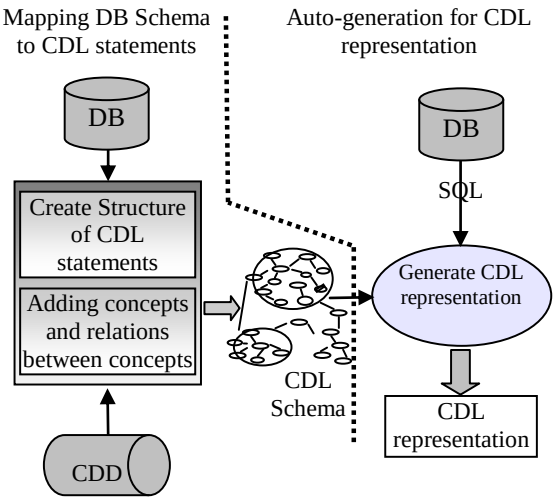


Figure 1. System architecture

CDL is one of the three forms that can be used to express CWL (Common Web Language). Moreover, CWL is a common language for exchanging information through the web and for enabling computers to process information semantically. CWL is a part of the Incubator Activity of W3C [14].

This new representation bases on Concept Description Language for natural language (CDL.nl) which describes the concept structure of the text based on a set of predefined semantic relations [12]. The main advantage of CDL is that it does not depend on ontologies. However, it depends on the Universal Networking Language Knowledge Based (UNLKB) and a set of universal relations so it can be used without facing similar problems of using ontologies. UNLKB is almost a complete dictionary, which contains around 120000 words. Moreover, this dictionary contains the definitions of these words represented into CDL form. This means that the computer can understand the word definition as well as the semantic relation between words that is also contained in that dictionary. Moreover, reasoning agents can use these definitions to get better understanding and more accurate results.

For example, representation of a statement such as "John bought a computer yesterday" in CDL looks like: {#A Event tmp='past';
{#a1 buy;}
{#a2 computer;}
{#a3 yesterday;}{John John;}
[#a1 agt John] [#a1 obj #a2] [#a1 tim #a3]}

III. RELATED WORK

RDB2RDF approach converts relational database to RDF in order to include this data into semantic web [5]. However, our approach converts database to semantic format that contains extra semantic relations to improve query answering process. Moreover, our proposed approach uses CDL which is richer than RDF because RDF represents data in triples format (object, property, value). However, CDL represents data in a semantic structure that enables users to express more internal relations. For example, RDF representation for the result of a query such as "select name, address from members where id =3;" is represented in two RDF triple as follows: (member, name, *Khaled*) (member, address, *Tokyo*). However, CDL representation for the same query result looks like (*Khaled*←agt–live—plc→*Tokyo*). There is an additional relation between name and address in CDL representation. This kind of relations is useful in answering queries semantically. Finally, the generated CDL does not depend on a domain ontology.

IV. PROPOSED SYSTEM ARCHITECTURE

The architecture of our system consists of two main components, DB mapping and CDL generator as shown in fig.1. The first step to represent DB into CDL is converting the DB schema to a semantic network. This semantic network is represented as a set of CDL statements, which contain references to DB objects. This process of converting DB schema to a semantic network occurs only once at installation time. The second step is auto-generation for the semantic of the DB content using those CDL statements.

A. *Representing DB schema into CDL semantic network*

In this stage, the schema of the DB is converted to semantic related concepts. The output of this process is a semantic network in which nodes represent concepts and arcs represent semantic relations between concepts. As shown in fig. 1, there are two steps for this conversion.

First, a semantic network structure that reflects database schema is created automatically by DB2CDL tool, fig. 2. In this step, for each data field in the DB a CDL statement is created.

Second, the user manually completes a CDL statement for each field by adding concepts and relations to the statement to make the overall meaning of the DB schema. For example, to

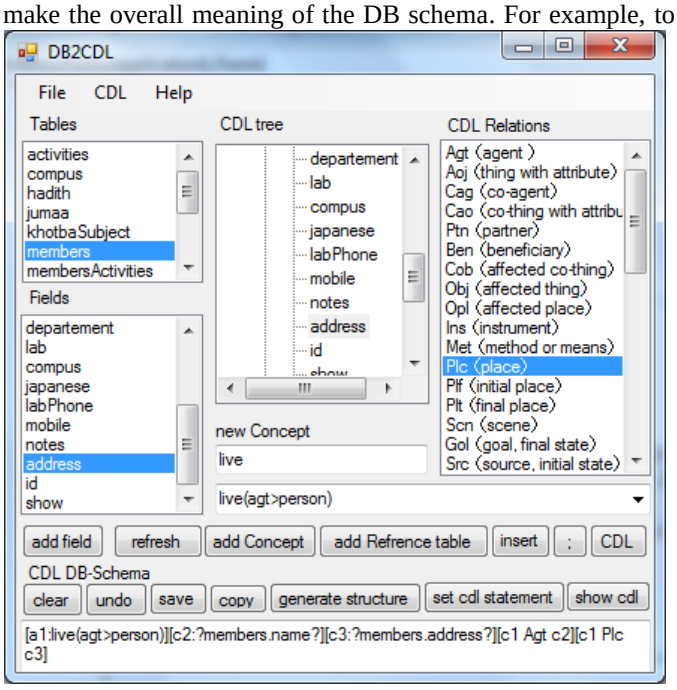


Figure2. DB2CDL tool

represent the field *address* in the table *members*, the user may insert a new event (concept) such as "*live*" and use CDL semantic relations to get the semantic representation for this field as the following notation: "M.name←agt live plc →M.address". This notation means that a member is an agent for the event *live* and the place of this event is the member's address. Moreover, when the user tries to add a concept to a CDL semantic network, the DB2CDL tool shows different usage forms of this concept according to CDD (Concept Definition Dictionary) of CDL language. The user should select one choice depending on the meaning. For example, if the user wants to add the verb live, he/she should select according to his meaning from the following alternatives.

live(agt>person,obj>thing)
live(agt>person,obj>food)
live(agt>person)
live(agt>thing,obj>state)
live(aoj>behavior)
…

In order to make the final generated semantic CDL accurate and reflects the overall meaning of the data, it is important to include DB relations into the outputted semantic network. Using the proposed tool (DB2CDL) the user can represent the DB field which represents a foreign key such as *deptID* field in *member* table, by creating a CDL statement which relates information from both tables (*member* and *department*). For instance, the user may add an event *"work"*, make a relation (*agent*) between this event and the *name* field in *member* table, and add another relation (*place*) between the new event and the *name* field in *department* table. Finally, the statement will be as the following:

```
<S field='deptID' ref='department'>
  <cdl>{ [a0:work(agt>person)]
  [a1:"?members.name?"]
  [a2:department(pof>organization)]
  [a4:"?department.name?"]
  [a1 agt a0][a2 plc a0][a4 mod a3] }::uw</cdl>
 </S>
```

The references to database objects in outputted CDL statements are replaced with the appropriate data in the CDL generation phase. Converting DB schema to CDL statements is not a time consuming or tedious work because the user does this only once. The proposed tool (DB2CDL), fig. 2, helps users to manage this conversion correctly and easily. Fig. 3 shows an example for the process of converting a DB table such as "member" table, which contains data of student society members, to CDL semantic statements.

The generated CDL statements will not be changed even though the stored data is changed. However, if the database schema changed, these CDL statements should be adapted to reflect schema changes.

B. CDL data generation

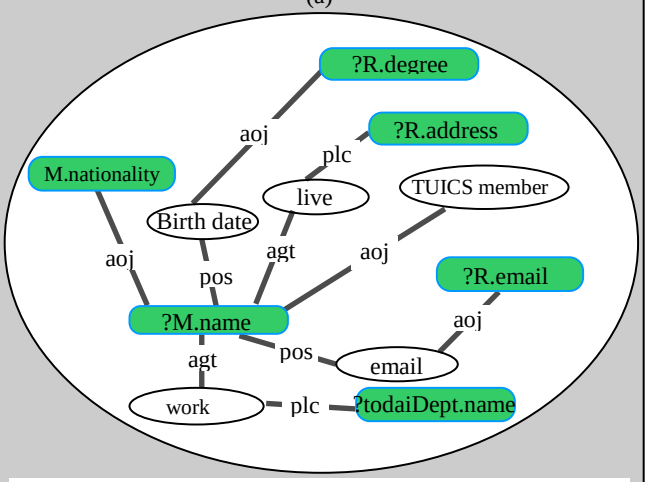
This stage automatically converts from relational DB to CDL format. This conversion is based on the semantic DB schema represented into CDL statements, which is the output of the first stage. This step is maintenance free. This means that there is no change in this step even though the database schema is changed.

In order to convert relational DB to CDL, first we run some basic SQL queries to retrieve all data from the DB. Queries results should be represented into CDL format to obtain the appropriate CDL data. In addition, the proposed technique provides converting all DB or a part of it to CDL.

```
<member>
<email>
 <cdl>[a7:email][a8:?members.email?][a9:?members.name?][a7 Aoj
      a8][a8 Pos a9]</cdl>
 </email>
<name>
 <cdl>[a5:TUICS member][a6:?members.name?][a5 Aoj a6]</cdl>
 </name>
<birthdate> <cdl>[b0:birthdate][b1:?members.birthdate?]
      [b2:?members.name?][b0 Aoj b1][b1 Pos b2]</cdl>
 </birthdate>
<Nationality>
 <cdl>[b3:?members.Nationality?][b4:?members.name?]
      [b3 Aoj b4]</cdl>
 </Nationality>
<department ref="todaiDept">
 <cdl>[b7:work(agt>person)][b8:?members.name?][c0:?
      todaiDept.name?]    [b7 Agt b8][b7 Plc c0]</cdl>
 </departement>
<address>
 <cdl>[b8:live][b9:?members.name?][c0:? members.address?]
      [b8 Agt b9][b8 Plc c0]</cdl>
 </address>
</member>
```

(b)

(a)



(a) a part of the semantic network of *members* table which contains members data (name, address, birthdate, email, …).
(b) an XML file that represent the CDL semantic network shown in (a)

Figure.3 CDL semantic schema example

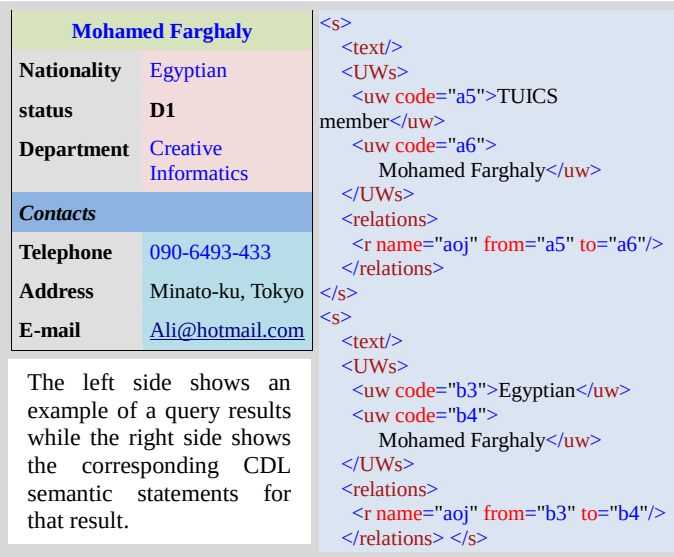


Figure 4. final CDL representation example

For example, consider an SQL query, *"Select member.name, phone, address, email, researchPoints, degree, department.name from member, department where member.id = Pr and member.deptID = department.ID;"*. If this query is run with the parameter Pr=42, it will show the information of a member with id = 42, fig. 4.

Based on semantic CDL schema that generated in the first phase we can generate CDL semantic of the query result. For example, it is stated in the CDL schema that the member name is the agent of an event '*live*' and the place of this event is the member's address. Consequently, by replacing DB references we obtain this CDL { live:0D >agt "Ali Saber":01; :0D >plc "Minato-ku, Tokyo":1D }::uw. Moreover, in order to auto-generate CDL data for query result the following steps should be executed.

1. get tables list stated in the *from* clause
2. get list of fields retrieved from each table
3. get CDL statement for each field from CDL semantic schema
4. for each field, generate the corresponding CDL statement by replacing DB references in the schema statement with values of query result.
5. find the DB relations stated in both *where* clause and CDL schema
6. for each field represent a relation (foreign key) find its CDL statement. If there is missing information for any statement, make a new query to get this information.

By applying these steps on the previous query, the representation of the query results will be as shown in fig. 4. These CDL statements contain semantics of the query results. Consequently, the generated data can be accessed semantically.

This phase of automatic CDL generation for query result was implemented as a set of APIs. These APIs can be used in different context. This means that CDL generation process can be run as a step of the execution of a program. As a result, web agents can easily consume the generated data and understand the content of the database.

Adding relations to the generated CDL representation

The generated CDL representation should contain rich semantic relations between different concepts that represent DB objects. The relations between DB objects are mapped to CDL. This work focuses on generating both relations (one-to-many and many-to-many) into the CDL format.

1- One-to-many relations

Normally, the CDL statement that attached to a foreign key field contains information from the reference table. For example, in DB schema shown in figure 5, the field *conference* in the *papers* table refers to *conferences* table. In the CDL semantic network for this DB schema, the CDL statement attached to *conference* filed contains information from both tables (*conferences* and *papers*). Consequently, a new query is created to retrieve values from the reference table. The new query is created automatically based on CDL semantic schema and the current record of the main table (*papers*). For instance, during converting *papers* table to CDL representation, for each record in *papers* table, we retrieve the related value from the *conference* table. As a result, the final CDL representation contains all relations of the DB.

2- Many-to-many relations

Representing many-to-many relation is quite similar to one-to-many relation. However, many-to-many relation is contained in a third table (bridge table). Therefore, we should check the related tables and include their statements into the generated CDL. Moreover, query creation to retrieve values of many-to-many relations is more complicated than one-to-many relation query creation. The following steps should be executed to include many-to-many relations into the CDL representation.

- For the current table find related bridge tables.
- Get CDL statements related to the current table
- Determine variables of these CDL statements
- Create query to retrieve values of the variables
- Run the query and replace variables by their values
- Include CDL to the final representation

V. EXPERIMENT

This section shows a prototype for the proposed approach. International Semantic Web Conferences (ISWC) DB was selected to implement the proposed approach. ISWC DB contains information about some conferences in semantic web field, published papers, authors, and so on. It contains information about 2600 authors and more than 1000 papers. The total numbers of records in this DB is 11213 records.

In order to represent ISWC DB contents into CDL, as a first step, DB schema (fig. 5) should be converted to CDL semantic network using DB2CDL tool. The resulted semantic network of ISWC DB stored in an XML file named *iswcCDLSchema.xml*. The second step is auto-generation of CDL statements that represents DB. The developed tool, DB2CDL generates the desired CDL data based on

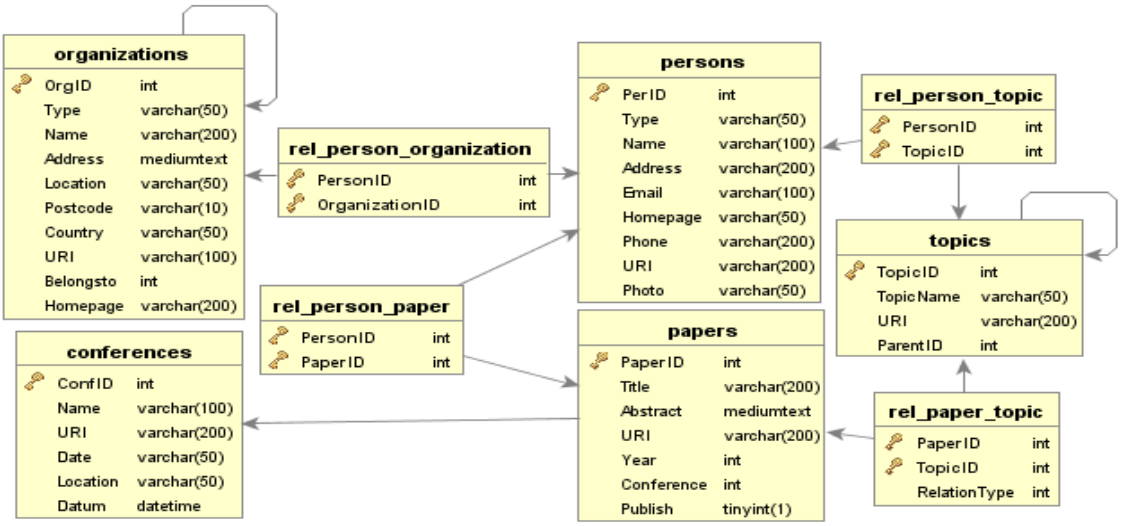


Fig.5 ISWC DB schema

iswcCDLSchema.xml file. The generated data contains semantic relations between different DB objects. The new semantic representation enables semantic agents to make the use of DB. The resulted CDL document contains 17657 CDL statements for the basic data (without DB relations). The generation time is 5.649 second. Full generation of CDL statement that represents data with all semantic relations takes 67.3 seconds and contains 52173 CDL statements.

Number of relations in the DB highly affects the execution time. This is because each relation needs to run a separated SQL query. Finally, the generated data represents all the relational DB into CDL format, which contains more semantic relations between concepts. Semantic web agents can use the generated data to get better interaction to web data.

VI. CONCLUSION

This paper proposes an approach to represent web DBs into a semantic format (CDL) which does not depend on ontologies. The proposed technique semi-automatically transforms the database schema to CDL semantic network using the implemented tool (DB2CDL). Based on generated CDL semantic network the proposed approach auto-generates semantic description of database content. The generated CDL enables intelligent agents to answer question and search the original content semantically. An experiment that converts large DB to CDL shows the visibility of the proposed approach.

REFERENCES

[1] T. Berners-Lee, J. Hendler, O. Lassila, “The Semantic Web”, Scientific American, Vol. 284, No. 5, 2001, pp. 34-43.

[2] Siegfried Handschuh , Raphael Volz , Steffen Staab, Annotation for the Deep Web, IEEE Intelligent Systems, v.18 n.5, September 2003, pp.42-48.

[3] Mamdouh Farouk, Samhaa R. El-Beltagy, Mahmoud Rafea, "On-the Fly Annotation of Dynamic Web ", Proceedings of the First International Conference on Web Information Systems and Technologies (WEBIST 2005)”, Miami (USA), may 2005, pp 327-332.

[4] Yulan Yan, Yutaka Matsuo, Mitsuru Ishizuka, Toshio Yokoi. "Annotating Extension layer of semantic structure for natural language text", The IEEE International conference on semantic computing. 2008, pp.174-181.

[5] Svihla, M., Jelinek, I.: The Database to RDF Mapping Model for an Easy Semantic Extending of Dynamic Web Sites. Proceedings of IADIS International Conference WWW/Internet, Lisbon, Portugal, 2005, pp.27-34

[6] Pan, Z. and Heflin, J.: DLDB: Extending Relational Databases to Support Semantic Web Queries, In Workshop on Practical and Scaleable Semantic Web Systems, The 2nd International Semantic Web Conference (ISWC2003) (2003).

[7] Ismael Navas Delgado, Nathalie Moreno Vergara, Antonio C. Gomez Lora, María del Mar Roldán García, Iván Ruiz Mostazo, José Francisco Aldana Montes: “Embedding Semantic Annotations into Dynamic Web Contents”. Proceeding of 15th international workshop on database and Expert Systems Applications, 2004, pp. 231-235

[8] Chris Bizer, and Richard Cyganiak :D2R server Publishing Relational Databases on the Semantic Web , www4.wiwiss.fu-berlin.de/bizer/d2r-server/ , 2010

[9] Zhuoming Xu, Shichao Zhang, and Yisheng Dong, Mapping between Relational Database Schema and Owl Ontology for Deep Annotation, WI'06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society, 2006, pp. 548-552.

[10] Ismael Navas Delgado, María del Mar Roldán García, José Francisco Aldana Montes: “Deep Crawling in the Semantic Web: In Search of Deep Knowledge”. WISE 2004, pp. 541-546

[11] Li Wenjie . Study of Semantic Web-Oriented Ontology Integration Technologies . In : Proceedings of the WRI World Congress on Software Engineering （WCSE'09） . Xiamen, China, May 2009, 2 : 142-145

[12] T. Yokoi, H. Uchida, K. Hasida, el al.CDL (Concept Description Language): A Common Language for Semantic Computing, www2005 workshop on the semantic computing initiative (SeC2005)

[13] Mitsuru Ishizuka, "A Common Concept Description of Natural Language Texts as the Foundation of Semantic Computing on the Web", IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Taiwan, June 2008, p.385

[14] H. Uchida, T. Yokoi, M. Zhu, N. Saito, V. Avetisyan, “Common Web Language”, W3C Incubator Group Report, http://www.w3.org/2005/Incubator/cwl/XGR-cwl/, March 2008