# Semantic Structure Content for Dynamic Web Pages

Mamdouh Farouk, Mitsuru Ishizuka
*Graduate School of Information Science & Technology, The University of Tokyo*
*mamdouh@mi.ci.i.u-tokyo.ac.jp, ishizuka@i.u-tokyo.ac.jp*

## Abstract

*Representing web data into a machine understandable format is a curtail task for the next generation of the web. Most of current web pages are dynamic pages. A large percentage of these web pages get their contents from underlying database. This work proposes an approach to represent dynamic web pages into Concept Description Language (CDL) semantic format. This format does not depend on ontologies which are domain dependant. However, CDL describes semantic structure of web content based on a set of semantic relations.*

## 1. Introduction

The main idea of the techniques that attempt to make web pages easy for semantically use is representing web data and resources in a standard semantic format to enable intelligent agents to interact web resources semantically and efficiently [1][2][3]. Related work uses semantic web languages (RDF, DAML, OWL,...), which depend on ontology, to represent meta-data that contains the semantics of the original data. However, this paper proposes a technique to represent dynamic web contents in a semantic structure format, CDL, that does not depend on ontologies.

The CDL (Concept Description Language), 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 [4]. The aims of CDL are to realize machine understandability of web text contents, and to overcome language barrier on the web.

CDL is one of three languages that can express CWL (Common Web Language). As a part of the Incubator Activity of W3C, CWL is a common language for exchanging information through the web and also for enabling computers to process information semantically.

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 [1]. The top ontology of CDL.nl is mainly based on the Universal networking Language Knowledge Based (UNLKB), which is based on Universal Words (UWs) of UNL developed under the United Nations (United Nations University). UWs is a large set of English words arranged in a specific semantic network structure.
The main advantage of CDL is that it does not depend on ontologies. However it depends on UNLKB and a set of universal relations so it can be used without ontologies problems. Moreover, the predefined CDL semantic relations can be universally use while ontologyies are domain dependant [5].

For example, representation of a statement such as "John bought a computer yesterday" in CDL is:

```
{#A Event tmp='past';
{#a1 buy;}
{#a2 computer ral='def';}
{#a3 yesterday ral='def';}
{John John;}
[#a1 agt John][#a1 obj #a2][#a1 tim #a3]}
```

```
//another form
{buy—agt→John;
buy—obj→computer;
buy—tim→yesterday;}
```

On the other hand, some researchers try to make representation to natural language text in web pages into CDL format [1]. This work uses natural language methods to make semantic representation. However, it is difficult to apply natural language techniques for tabular form in dynamic web pages that get its content from underlying database. This is because tabular form does not follow the natural language rules. However, we adopt another approach to represent this data form in CDL language. This paper focuses on representing dynamic web content that are generated from underlying database into CDL semantic format. The generated CDL semantic representation can be used to make intelligent search over a wide range of web pages to get accurate search results. Actually, searching semantic CDL relation is more powerful than searching relational database or natural language text in the web pages.

978-0-7695-4191-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WI-IAT.2010.290

## 2. Architecture

This work aims to represent dynamic generated date, stored in a database, into CDL format in order to be processed semantically. This approach depends on the underlying database schema. This is because web page design is more likely to change than the

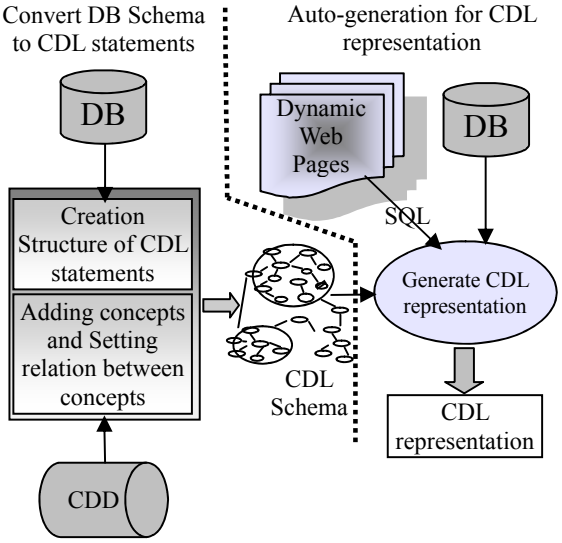



Figure 1. System architecture

underlying stored data structure. Consequently, CDL generation will not change even web page design is changed.

In order to represent the meaning of dynamic web pages, as a first step the underlying DB schema should be converted to a semantic network represented as a set of CDL statements. These statements contain references to DB objects. This process of representing DB schema in a semantic network should occur only once at installation time. Moreover, the semantic of the webpage content can be auto-generated using these CDL statements. As shown in figure 1 this work is divided into two main phases. The first phase is to make representation to relational database schema in CDL notation. The second phase is auto-generation for CDL semantic representation for dynamic web pages.

## 3. DB schema semantic representation

The aim of this stage is to convert DB objects (tables and fields) and the relations between these objects into semantic related concepts represented by CDL format. The output of this process is a semantic network in which nodes represent concepts and arcs represent semantic relations between concepts. As shown in figure 1, there are two steps to make this conversion.

The first step to convert DB schema into CDL representation is automatically creation of a semantic network structure that represents database schema. In this step, a paragraph is created for each database table and for each data field in this table a statement inside the paragraph is created.

Moreover, in this automatically generated template for semantic network structure, table name appears as a property in the paragraph tag (<P>). In addition, each statement inside the paragraph is associated to a specific table's field through *field* property. The *ref* property is used to indicate that there is a relation between current table and this table which stated as a value of *ref* property. The *ref* property, which is used to generate more accurate CDL representation, appears in the statement associated to a foreign key field.

As a second step, user should manually complete a CDL statement for each field by adding missing concepts and relations to the statement. For example, to represent researcher *address* field, user may insert a new event such as "*live*" and make the use of CDL semantic relations to get the semantic representation for this field as the following notation: "R.name←agt live plc →R.address". This notation means that a research is an agent of the event live and the place of this event is the researcher address.

In addition, to represent the statement that associated to a foreign key such as *deptID* field, user

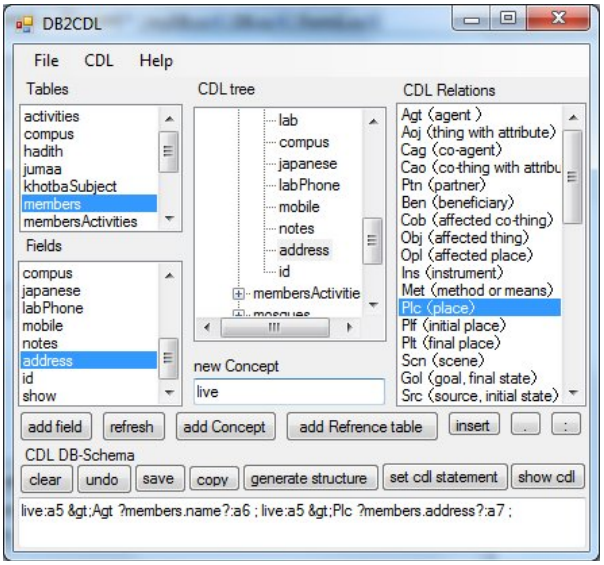



Figure2. DB2CDL tool

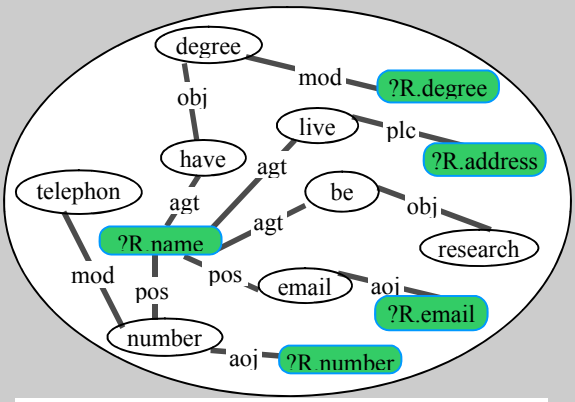
may add an event *"work"* , make a relation between this event and the *name* field in researcher table, and add another relation between the new added event and the *name* field in department table. Finally, the statement will be as the following:

```
<S field='deptID' ref='department'>
  <cdl>{ work:0A >agt "?researcher.name":01,
>plc depatment:3A;
  :3A >mod "?department.name":4A }::uw</cdl>
 </S>
```

```
<researcher>
  <name>
    <cdl>be:b1 >Agt ?researcher.name?:b2 ;be:b1 >Obj
researcher:a3 </cdl>   </name>
  <address>
    <cdl>live:b3 >Agt ?researcher.name?:b4 ;live:b3 >Plc
?researcher.address:b5 </cdl>   </address>
  <phone>
    <cdl>?researcher.phone?:b6 >Aoj number:b8 >Pos
?researcher.name:b9;  number:b8 >Mod telephone:c0 </cdl>
  </phone>
  <email>
    <cdl>?researcher.email?:c1 >Aoj email:c2 >Pos
?researcher.name:c4 </cdl>   </email>
  <degree>
    <cdl>have:c5 >Agt ?researcher.name?:c6 ;have:c5 >Obj
degree:c7;  degree:c7 >Mod ?researcher.degree?:c8 </cdl>
  </degree>
</researcher>
```

(b)

(a)




(a) a part of the semantic network of researcher table which contains researchers data (name, address, number, email, deptID, degree, researchPoints) while (b) the code that represent the CDL semantic network shown in (a)

Figure.3 CDL semantic schema

The references to database objects are replaced with the appropriate data in the CDL generation phase. Completing CDL statements for DB schema is not a time consuming or tedious work because user works only on DB schema not the actual data. The proposed (DB2CDL) tool, figure 2, helps users to manage conversion of DB schema to CDL semantic network correctly and easily. Figure 3 shows an example for the process of representing a db table such as "*researcher*" table in the proposed semantic form.

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

## 4. CDL data generation

This stage automatically converts dynamic web page that retrieves its contents from DB to CDL format. In order to make this conversion it should use the semantic DB schema represented in 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 stored data is changed or the database schema is changed.

Dynamic web page that retrieves its contents from underlying database contains SQL queries that are executed on server side to generate the page content. In order to represent the page content in semantic format, query result should be represented in the CDL format.

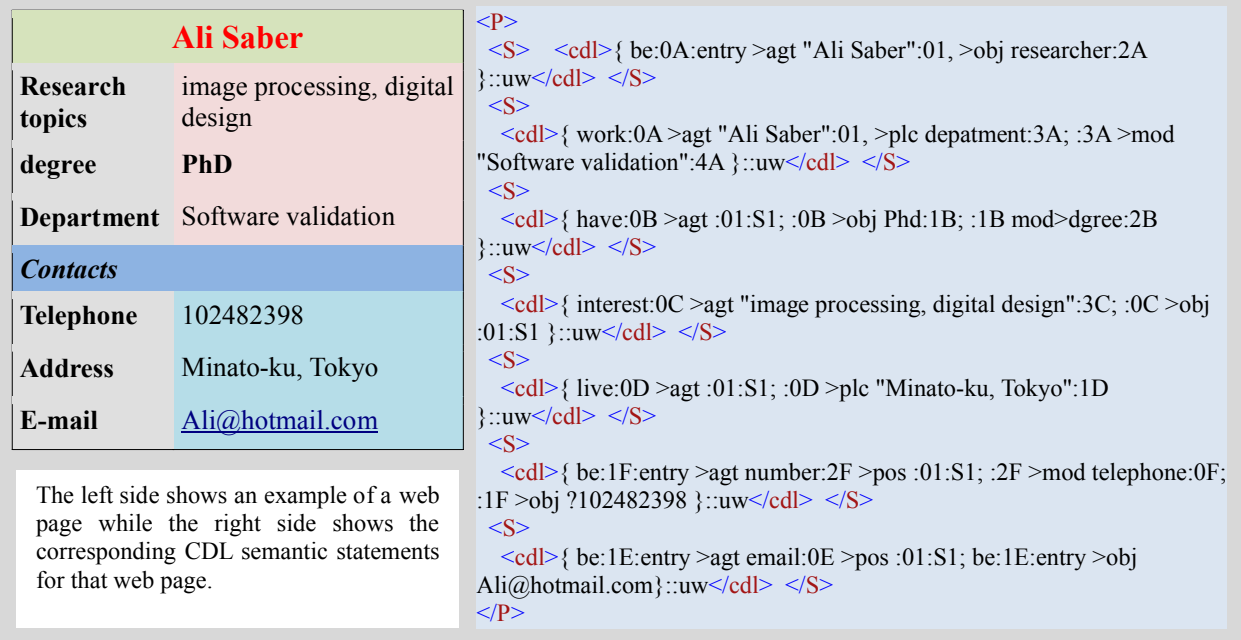
For example, consider a dynamic web page, that shows researchers information, contains this SQL query *"Select researcher.name, phone, address, email, researchPoints, degree, department.name from Researcher, department where researcher.id = Pr and researcher.deptID = department.ID;"*. If the web server receives a request to this page with parameter Pr=42, it will show the information of a researcher with specific id = 42, figure 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 DB-schema that the researcher name is the agent of an event 'live' and the place of this event is the address. consequently, 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 CDL schema for each table (table's paragraph)
3. get list of fields retrieved from each table
4. find the DB relations stated in both *where* clause and CDL schema
5. for each field generate the corresponding CDL statement (replacing DB references)
6. for each relation 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 web page contents will be as shown in figure 4. These CDL statements contain semantic of the web page contents. Consequently, this dynamic web page can be accessed semantically.

This phase of automatic CDL generation for dynamic web page contents can be implemented as a set of APIs. These APIs can be used in the script of the web page. This means that if a request is received by the web server to a specific page the CDL generation process will be run as a step of script execution on the server side and the generated CDL code will be sent to



| Ali Saber | |
|---|---|
| **Research topics** | image processing, digital design |
| **degree** | **PhD** |
| **Department** | Software validation |
| ***Contacts*** | |
| **Telephone** | 102482398 |
| **Address** | Minato-ku, Tokyo |
| **E-mail** | Ali@hotmail.com |

The left side shows an example of a web page while the right side shows the corresponding CDL semantic statements for that web page.

```
<P>
 <S>  <cdl>{ be:0A:entry >agt "Ali Saber":01, >obj researcher:2A
}::uw</cdl> </S>
 <S>
  <cdl>{ work:0A >agt "Ali Saber":01, >plc depatment:3A; :3A >mod
"Software validation":4A }::uw</cdl> </S>
 <S>
  <cdl>{ have:0B >agt :01:S1; :0B >obj Phd:1B; :1B mod>dgree:2B
}::uw</cdl> </S>
 <S>
  <cdl>{ interest:0C >agt "image processing, digital design":3C; :0C >obj
:01:S1 }::uw</cdl> </S>
 <S>
  <cdl>{ live:0D >agt :01:S1; :0D >plc "Minato-ku, Tokyo":1D
}::uw</cdl> </S>
 <S>
  <cdl>{ be:1F:entry >agt number:2F >pos :01:S1; :2F >mod telephone:0F;
:1F >obj ?102482398 }::uw</cdl> </S>
 <S>
  <cdl>{ be:1E:entry >agt email:0E >pos :01:S1; be:1E:entry >obj
Ali@hotmail.com}::uw</cdl> </S>
</P>
```

Figure 4. CDL example

the client as a part of page content. As a result, web agents can easily understand this kind of web pages.

## 5. Related work

Comparing to RDB2RDF approach, which converts relational database to RDF [6], CDL is richer than RDF. This is 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: (member, name, Khaled) (member, address, Tokyo). However, CDL representation for this query result looks like (Khaled←agt–live—plc→Tokyo). So in CDL there is additional relation between name and address. This kind of relations is useful in answering queries semantically. Finally, unlike RDF, CDL does not depend on domain ontology.

## 6. Conclusion

This paper shows a technique to represent dynamic web pages into CDL (Concept Description Language). The proposed technique transforms the database schema to CDL semantic network structure (which is a model transformation step) and manually completes CDL statements with semantic relations using DB2CDL tool. As a second step, extract semantic description associated to a page according to the retrieved data it contains. This last step is based on SQL queries used to bring contain into dynamic pages.

## 7. References


[1] 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.

[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] 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)

[5] 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

[6]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