Ontology based Web Application Reverse-Engineering Approach

BOUCHIHA Djelloul [1], MALKI Mimoun [2], BENSLIMANE Sidi Mohamed [3]

[1] EEDIS Laboratory, University of Sidi Bel Abbes 22000, Algeria
bou_dje@yahoo.fr
[2] EEDIS Laboratory, University of Sidi Bel Abbes 22000, Algeria
Malki_m@yahoo.com
[3] EEDIS Laboratory, University of Sidi Bel Abbes 22000, Algeria
benslimane@univ-sba.dz

Abstract. With the Web's emergence and generalization in various domains such as economy, commerce, education, culture, etc, the Web application reverse-engineering process becomes necessary in order to facilitate the maintenance of such applications and the evolution towards new Web technology like XML, semantic Web, etc. In this paper, we propose a new approach for the Web application reverse-engineering. The approach is based on ontology and it generates a conceptual schema modelling the Web application. This conceptual schema is rich in semantic but reduced in relation to the global ontology. The proposed approach mainly relies on HTML pages analysis, i.e. to analyse tables, lists, forms, etc. It consists of three successive phases: First, the extraction of useful information from the HTML pages. Second phase is the analysis of the extracted information using the domain ontology. And finally, we generate the corresponding UML conceptual schema.

Keywords

Reverse-engineering, Maintenance, Evolution, Web Application, Ontology, Semantic distance, UML.

(Received May 27, 2006 / Accepted November 10, 2006)

1. Introduction

With the advent of Internet and Web-enabled technologies, new applications are developed in the format of software systems that deliver specific functionalities as services through the Web. These applications are required to undergo a reverse engineering process to be maintained, evolved, migrated, and understood.

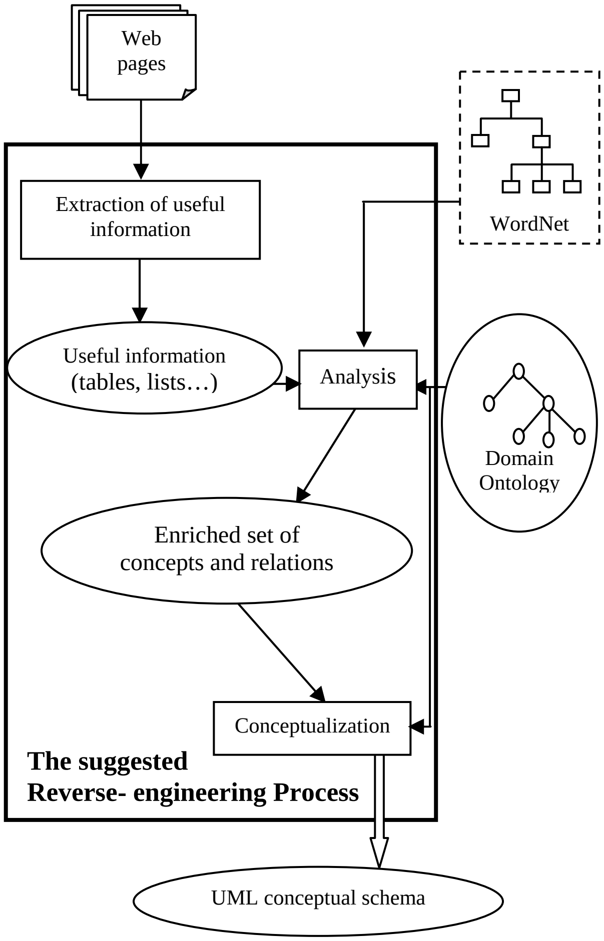
Several reverse-engineering approaches were proposed. These approaches aim a conceptual schema as a target schema [2], [5], [6], [7], [8], [9], [4], [24], [25].

For their objective, the majority of Web application reverse-engineering approaches adapted techniques of data-bases reverse-engineering. For us, we tried to adapt the approach of MALKI and al [14], [15], [16], [17], [18], who proposed a bottom-up/top-down hybrid approach based on forms analysis to perform the data bases reverse-engineering.

All the Web application reverse-engineering approaches follow a strictly bottom-up abstraction process, i.e. abstraction is obtained through transformation of the analyzed logic schema. This bottom-up approach is less adequate if a preexistent partial design for data structure (documentation or knowledge of designer or domain expert) and/or domain ontology of the application are available. For that, we propose a Web reverse-engineering approach requiring a bottom-up/top-down hybrid data abstraction process. This approach consists in analyzing dynamic Web pages while being centred on tables, lists, forms, etc. It is based on domain ontology for generating a conceptual schema.

2. Ontology Based Web Application Reverse-Engineering Approach

The approach consists in three successive phases (Fig. 1.):



Web pages
WordNet
Extraction of useful information
Useful information (tables, lists…)
Analysis
Domain Ontology
Enriched set of concepts and relations
Conceptualization
The suggested Reverse- engineering Process
UML conceptual schema

Fig. 1. Ontology based Web Application Reverse-engineering Process.

First phase is the Extraction of useful information from HTML pages. Useful information is this presented in tables, lists, forms, etc. Second phase is the Analysis which stars with a comparison accomplished between the extracted information of the preceding phase and the concepts of ontology by using semantic distance techniques. This semantic comparison permits the extraction of a concept set which presents the content of HTML pages. Then, the inference of new concepts and relations is done. The last phase consists in generating an UML conceptual schema.

The objective of our approach is to extract a conceptual schema describing the Web application based on domain ontology.

Initially, we assume that:

1. In HTML pages, a concept of domain ontology can be indicated by a table field or a list element; Or else, it can be indicated by a set of table fields or a set of list elements.
2. An ontology exists, built by experts and it is specific to a domain (domain to which belong the HTML pages).
3. From the domain ontology, we can extract the global conceptual schema describing the entire domain.

Next, we present in detail the phases of the ontology based Web application reverse-engineering process.

2.1. Extraction of useful information

This phase starts with filtering HTML pages, followed by the extraction of DOM[1] and finally the extraction of useful information from DOM (Fig. 2.). Filtering consists in browsing the source code of HTML pages, eliminate useless tags such as those of layout (e.g. <b>, <i>), and preserve useful tags, which carry information to be treated in the following stages (e.g. <form>,

[1] DOM : Document Object Model is an API witch consists in decomposing the HTML or XML document content in a tree structure of nodes (each element of the document is a node).

<table>, <td>, <tr>, <ul>, <li>). The result of this step is a set of cleaned HTML pages.

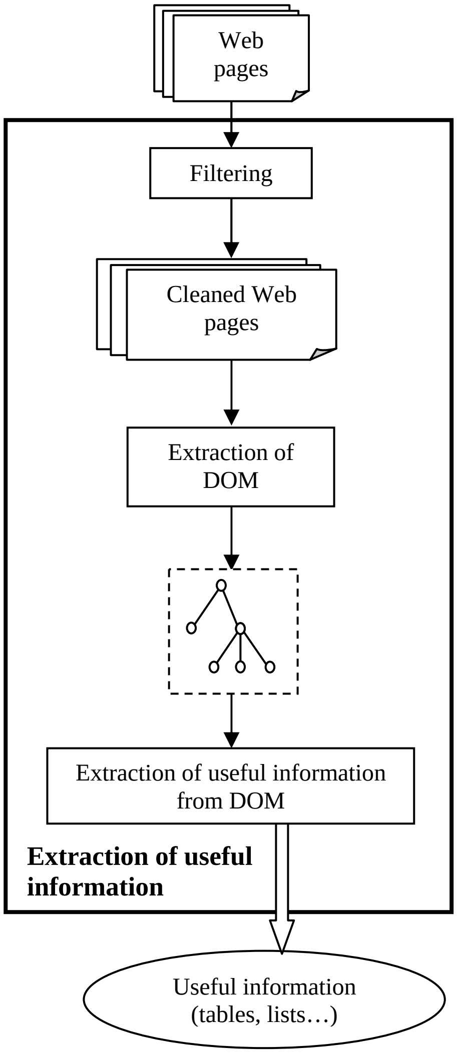

Web pages
Filtering
Cleaned Web pages
Extraction of DOM
Extraction of useful information from DOM
Extraction of useful information
Useful information (tables, lists…)

Fig. 2. The phase of Extraction of useful information.

Cleaned HTML pages will be presented in DOM logical format in order to facilitate there manipulation. From this DOM logical format, we can now extract the useful information stored in tables, lists, forms etc.

2.2. Analysis

The analysis phase is a steps series for the treatment of useful information resulting from the previous phase. The result of this phase is the generated UML conceptual schema describing the Web application (Fig. 3.).

Morphological analysis. The analysis phase starts with a morphological analysis applied to the tables fields and the lists elements extracted from HTML pages. The morphological analysis consists in removing hyphens and keep terms stem as they appear in WordNet[2] (e.g. morphological analysis of ‘running-away’ is ‘run away’). The result of this stage is a set of terms which can be identified after as concepts of ontology.

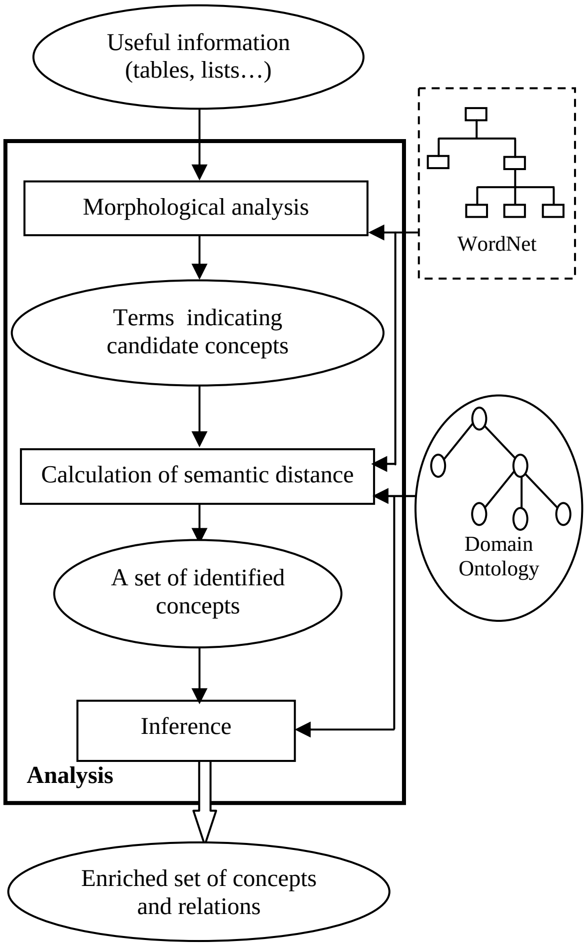

Useful information (tables, lists…)
Morphological analysis
WordNet
Terms indicating candidate concepts
Calculation of semantic distance
Domain Ontology
A set of identified concepts
Inference
Analysis
Enriched set of concepts and relations

Fig. 3. Analysis phase.

Calculation of semantic distance. Several approaches have been proposed to calculate semantic distance between two concepts; those approaches aim to quantify how much two concepts are alike.

[2] WordNet is a lexical data base witch organizes names and verbs in concepts (synset) in is-a hierarchy of relations. Each concept is described by a short gloss.

Semantic distance approaches relying on WordNet can be classified into three categories:

1. Similarity measures based on path lengths between concepts: The *LCH* [12], the *WUP* [23] and the *path* measure. The last one is equal to the inverse of the shortest path length between two concepts.

2. Similarity measures based on information content: The *RES* [21], the *LIN* [13] and the *JCN* [11] measure.

3. Relatedness measures based on relations type between concepts: The *HSO* [10], the *LESK* [3] and the *VECTOR* [20] measure.

To calculate semantic distance between two groups of entities, several strategies are used. Some of these strategies are used for the hierarchical clustering [1].

- *Single linkage:* Distance between groups is defined as the distance between the closest pair of objects.

- *Complete linkage:* Distance between groups is now defined as the distance between the most distant pair of objects.

- *Average linkage:* Distance between two clusters is defined as the average of distances between all pairs of objects.

Another strategy used for calculates similarity between sets in ontologies [19].

- *Multidimensional scaling:* It is a statistical technique for the calculation of the semantic distance between two sets of entity. The formula of calculation is as follows :

$$sim_{set}(E,F) := \frac{\sum_{e \in E} e}{|E|} \cdot \frac{\sum_{f \in F} f}{|F|},$$

with entity set $E = \{e_1, e_2, ..\}$,

$$e = (sim(e,e_1), sim(e,e_2), ...sim(e, f_1), ...);$$

F and *f* are defined analogously.

Methods and strategies described above will be used in the stage of calculation of semantic distance in our retro-engineering process as follows: We have each table or list corresponds to an anonymous entity, described by its fields or its elements. We have also a domain ontology containing concepts; where each concept has possibly a set of attributes. Now we proceed to calculations of semantic distance to identify the concepts of ontology hidden in HTML pages.

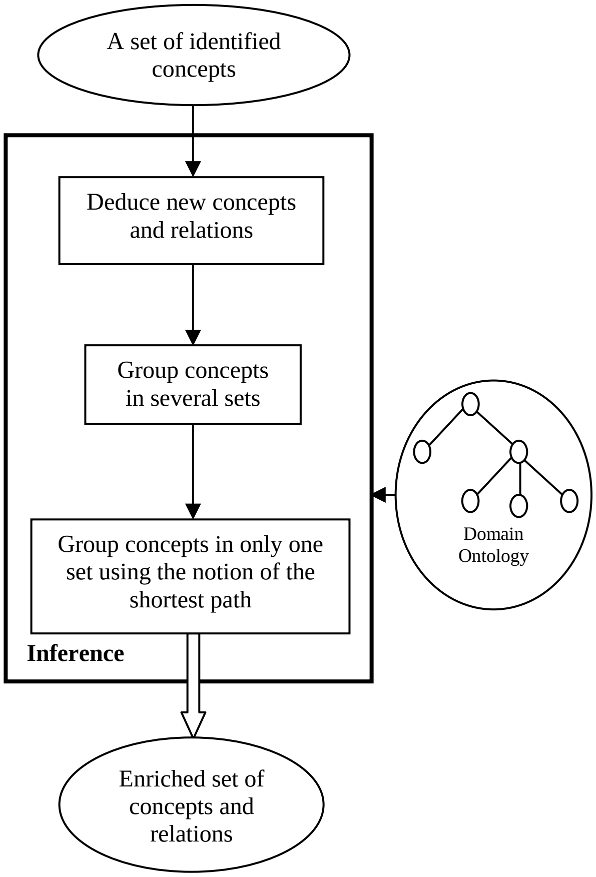
For that we must fix a threshold for the semantic distance. The threshold is a value between 0 and 1, the 1 value indicates that the two entities are totally similar. Then we must choose a method and a strategy to calculate semantic distance as described above.

If the semantic distance between a table field (or a list element) in the HTML pages and the name of a concept of the ontology is superior or equal than the threshold fixed before, then we consider that the concept of ontology is identified and it is marked as an existing concept.

Moreover, if the semantic distance between two entities groups is superior or equal than the threshold, then we consider that the concept of domain ontology corresponding to one group is identified. Each two groups are constructed as follow: the first one is the table fields (or the list elements) in the HTML pages, the second is the concept attributes of the domain ontology.

Inference. Inference consists in inferring new concepts and relations before generating the UML conceptual schema describing the Web application based on ontology (Fig. 4.).

The inference starts with the deduction of new concepts and relations. We can enrich our target conceptual schema by determining the concepts in relation with the first concepts identified after the stage of calculation of semantic distance. That is done by using -of course- the ontology. In other words, each relation witch has on a side an identified concept; it must appear in the conceptual schema, as well as all the concepts on its two sides. The objective of this stage is the enrichment of the resulting conceptual schema.



A set of identified concepts
Deduce new concepts and relations
Group concepts in several sets
Group concepts in only one set using the notion of the shortest path
Inference
Domain Ontology
Enriched set of concepts and relations

Fig. 4. Inference process.

Now we will obtain a set of concepts and relations with which we can form a set of groups. Each group represents a connected graph[3].

Before generating the final conceptual schema, the resulting groups from the previous stage must be unified in a single group, without congesting the schema by several other concepts and relations. Each two groups can be unified by the shortest path between concepts of both groups in the hierarchy of ontology.

2.3. Conceptualization

From the Enriched set of concepts and relations extracted from the previous phase we can construct an UML conceptual schema (Fig. 5.) as follow: each concept and relation of the extracted set will be presented respectively by an UML class and relation in the resulting schema. The relation expressed by the term 'part-of' will be presented as an UML aggregation relation.

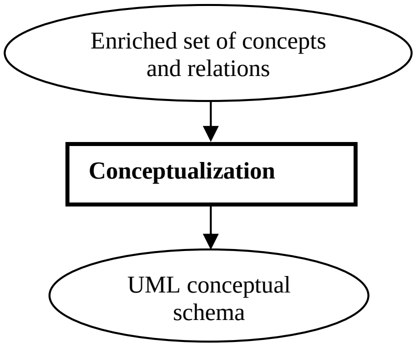


Fig. 5. Conceptualization phase.

The subsumption relations which appear in the shortest paths during the unification of groups in only one set will be translated by an heritage link in the conceptual schema. Even the multiple heritage can appears in this schema.

Cardinalities relations are also extracted from domain ontology to be presented in the UML conceptual schema.

3. Implementation

To implement our ontology based Web application reverse-engineering approach, we have developed a tool named OntoWeR (Ontology based Web Reverse-engineering). The system receives as input a well formed HTML pages[4] + ontology expressed in OWL-DL. Then the system applies a description logics based reasoning by using the Protege-OWL API[5].

[3] A graph is connected if and only if there is a path between any pair of vertex in the graph.

[4] An HTML page is well formed if all tags have an ending tag or are themselves self-ending.

[5] The Protege-OWL API is an open-source Java library for the Web Ontology Language (OWL) and RDF(S).

To evaluate the Ontology based Web application reverse-engineering approach -proposed in this paper-, we are going to present a case study. For that, we used a preexistent ontology and a Web site on which we are going to do experimentation. At the end, we will discuss the obtained results.

3.1. Architecture of the implemented system

The implemented tool is composed of four subsystems to allow assuming the stages of Web application reverse-engineering process, as well as ensuring interaction with the user (Fig. 6.):

- Acquisition Module: allows the acquisition of HTML pages of Web application, as well as domain ontology. It allows user to fix semantic distance threshold and to choose method and strategy for calculating this distance.

- Extractor: represent an implementation of the extraction phase in the reverse-engineering process. It allows extracting useful information from the acquired HTML pages.

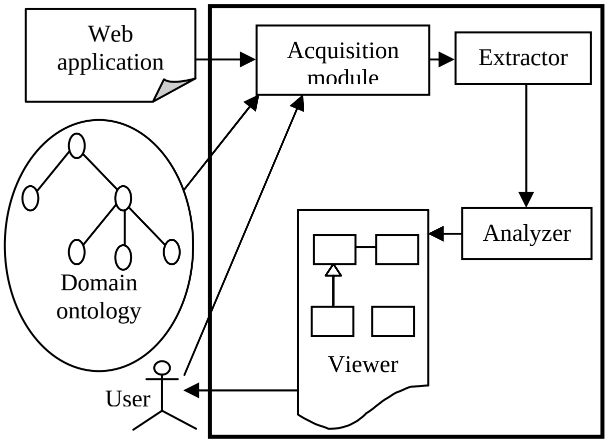


Fig. 6. Architecture of the implemented system.

- Analyzer: represent an implementation of the analysis phase in the reverse-engineering process. It covers calculation of semantic distance, deduction of new concepts and relations, and inference, aiming to generate an UML conceptual schema.

- Viewer: allows viewing the resulting conceptual schema. It also allows viewing a final report detailing all calculations and operations performed in progress of the reverse-engineering process.

3.2. The tourism domain ontology

The chosen ontology is a tutorial ontology for a Semantic Web of tourism[6]. It describes the tourism domain. It is contributed by Mr Holger Knublauch[7]. The next is an extract from this ontology.

```
….
 <owl:Class rdf:ID="Hotel">
   <owl:disjointWith>
     <owl:Class rdf:about="#BedAndBreakfast"/>
   </owl:disjointWith>    <owl:disjointWith>
     <owl:Class rdf:about="#Campground"/>
   </owl:disjointWith>
   <rdfs:subClassOf rdf:resource="#Accommodation"/>
 </owl:Class>
 <owl:Class rdf:ID="Museums">
   <rdfs:subClassOf>
<owl:Class rdf:about="#Sightseeing"/>
   </rdfs:subClassOf>
 </owl:Class>
…
```

3.3. Presentation of the Web site

The chosen Web site on which we perform our experiments is http://www.hm-usa.com/. It is a Web site for tourism in the United States of America.

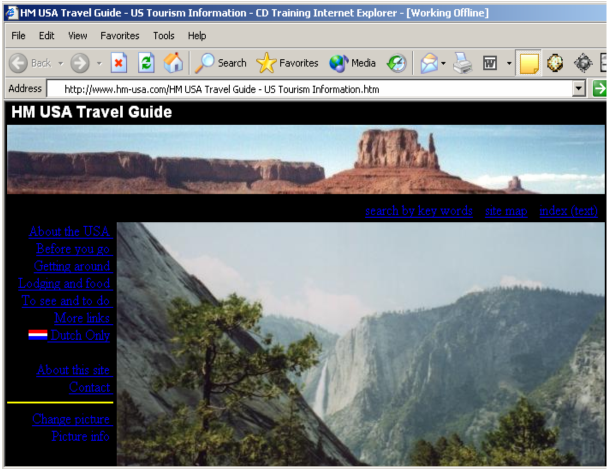
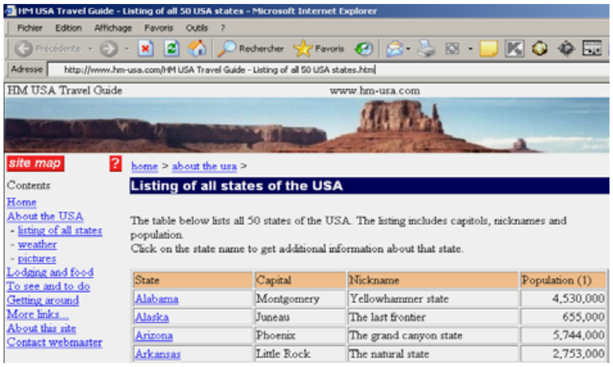


Fig. 7. Home page of the site.

[6] http://protege.stanford.edu/plugins/owl/owl-library/travel.owl

[7] http://www.knublauch.com/

From this Web site, we choose two HTML pages. The first page presents the states of USA; the second one presents the hotels.


HM USA Travel Guide - Listing of all 50 USA states - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente
Rechercher
Favoris
Adresse
site map
?

http://www.hm-usa.com/HM USA Travel Guide - Listing of all 50 USA states.htm

HM USA Travel Guide www.hm-usa.com

home > about the usa >

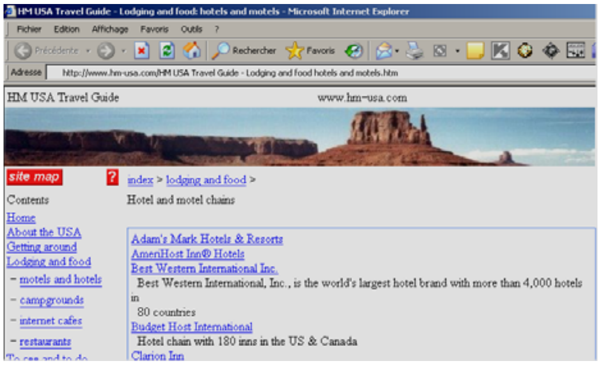
Contents
Home
About the USA
- listing of all states
- weather
- pictures
Lodging and food
To see and to do
Getting around
More links...
About this site
Contact webmaster

Listing of all states of the USA

The table below lists all 50 states of the USA. The listing includes capitols, nicknames and population.
Click on the state name to get additional information about that state.

State	Capital	Nickname	Population (1)
Alabama	Montgomery	Yellowhammer state	4,530,000
Alaska	Juneau	The last frontier	655,000
Arizona	Phoenix	The grand canyon state	5,744,000
Arkansas	Little Rock	The natural state	2,753,000

Fig. 8. Page of USA's states.


HM USA Travel Guide - Lodging and food: hotels and motels - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente
Rechercher
Favoris
Adresse
site map
?

http://www.hm-usa.com/HM USA Travel Guide - Lodging and food hotels and motels.htm

HM USA Travel Guide www.hm-usa.com

index > lodging and food >

Contents
Home
About the USA
Getting around
Lodging and food
- motels and hotels
- campgrounds
- internet cafes
- restaurants
To see and to do

Hotel and motel chains

Adam's Mark Hotels & Resorts
AmeriHost Inn® Hotels
Best Western International Inc.
Best Western International, Inc., is the world's largest hotel brand with more than 4,000 hotels in
80 countries
Budget Host International
Hotel chain with 180 inns in the US & Canada
Clarion Inn

Fig. 9. Page of USA's hotels.

3.4. UML conceptual schema

With a threshold of semantic distance (using *path* measure) equal to 0.7 and by choosing the *Multidimensional scaling* strategy to calculate the distance between groups. The following conceptual schema is obtained:

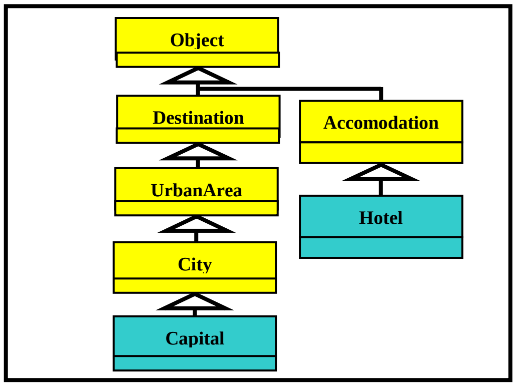


Fig. 10. Conceptual schema in the first case.

With a threshold of semantic distance (using *path* measure) equal to 0.1 and by choosing the *Multidimensional scaling* strategy to calculate the distance between groups. The following conceptual schema is obtained:

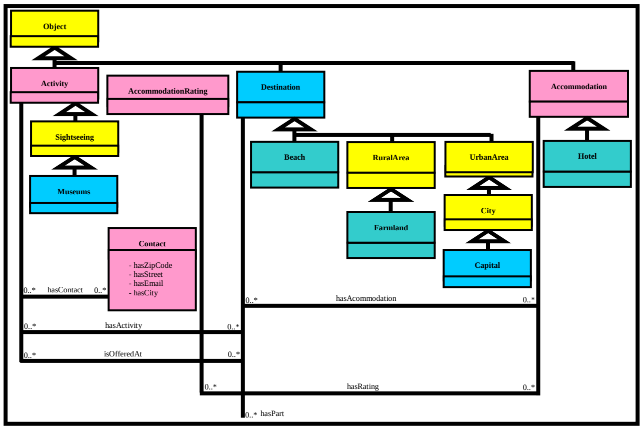


Fig. 11. Conceptual schema in the second case.

3.5. Discussion

The system generates a colored conceptual schema. The signification of each color in the resulting conceptual schema is as bellow:

– The colored classes in blue are those identified after the stage of calculation of semantic distance.

– The colored classes in chestnut are those appeared after the stage of deduction.

– The colored classes in yellow are the classes obtained during the grouping of sets (by the shortest paths).

In addition to UML conceptual schema, the system offered other services. We can personalize the schema (Dimensions, writing size). We can also visualize the details on operations and calculations performed during the reverse-engineering process.

The unique method of semantic distance implemented in our system is the path method. For calculating the semantic distance between groups, we implemented the four strategies described above. After several tests, we noted that:

- For the *path* measure, more the semantic distance threshold is small; more the conceptual schema becomes complex and vise versa.
- The *Multidimensional scaling* strategy gives more efficient results than the other strategies.

The current system represents only static aspect of the Web application. It can be extend to represent dynamic aspect.

By using WordNet we can analyze only English sites. This problem can be solved by the use of a multilingual lexical data base.

Fixing a threshold for semantic distance means that an error rate was tolerated. We tolerate this risk because the conception domain is not deterministic, but it is heuristic. There is not a unique correct model for a situation, only adequate or inadequate models [22]. Also, the implication of ontology makes the results much more adequate to the application field. To avoid any ambiguity and get more realistic result, we fix the threshold at 1.

The strong point of our approach is that it relies on a very rich semantic reference which is the ontology. We can affirm that our approach gives very satisfactory results which can be used for reengineering, migration, understanding, and evolution of the Web applications.

4. Conclusion

The purpose of this paper is to propose an approach for reverse-engineering Web applications using domain ontology to generate conceptual schemas, for these applications, expressed in UML.

The proposed reverse-engineering process consists in three phases: Analyses, Extraction of useful information, and Conceptualization. The approach allows users to recover static and dynamic Web sites.

In our approach we consider ontology as the main semantic source which permits the identification of the hidden concepts in HTML pages.

To implement our ontology based Web application reverse-engineering approach, we developed a tool named OntoWeR (Ontology based Web Reverse-engineering). Our system can be adopted to extract another type of information, other than conceptual schema, since it relies on a very rich semantic source which is -of course- the ontology.

In our approach, we supposed that from ontology we can extract a conceptual schema describing the domain of ontology. That recall us another research axis witch is the reverse-engineering of ontology itself. Therefore our system can be adapted to such work.

Web services are a new technology that has a lot importance. They are software components encapsulating functionalities of enterprise, and accessible via standard protocols. These Web services can call a phase of reverse-engineering to allow their evolution. For that we think of extending our approach toward the Web services reverse-engineering and the migration of Web applications toward Web services.

References

[1] Alexander M, *ONTOLOGY LEARNING FOR THE SEMANTIC WEB*. Un ouvrage distribute par Kluwer Academic Publishers. 2002.

[2] Antoniol G., Canfora G., Casazza G., and De Lucia A., *Web Site Reengineering Using RMM*. 2nd nd International Workshop on Web Site Evolution. March 1, 2000.

[3] Banerjee, S., and Pedersen, T., Extended gloss *overlaps as a measure of semantic relatedness*. In Proceedings of the Eighteenth International Joint Conference on Artifi-cial Intelligence, Pages 805–810. 2003.

[4] Filippo Ricca and Paolo Tonella. *Web application slicing*. Proceeding of the international conference on software maintenance. Page 148-157. Firenze, Italy 2004.

[5] Chung S., Lee Y.(2000), *Reverse Software Engineering with UML for Web Site Maintenance*. Proceedings of the First International Conference on Web Information Systems Engineering (WISE'00)-Volume2-Page: 2157. 2000.

[6] Di Lucca G.A., Di Penta M., Antoniol G., Casazza G., *An Approach for Reverse Engineering of Web-Based Applications*. Proceedings of the Eighth Working Conference on Reverse Engineering. Pages: 231-240. Stuttgart, Germany, Oct 2001.

[7] Fabrice E., Aurore F., Jean H., Jean-Luc H., *A tool-supported method to extract data and schema from web sites*. Proceedings of the fifth international workshop on Web site evolution. Pages: 3-11. Amsterdam, 2003.

[8] Filippo R., Paolo T., *Analysis and Testing of Web Applications*. In Proc. Of ICSE’2001, Int. Conf. on Soft. Eng. Pages: 25-34, Toronto, Canada, May 12-19, 2001.

[9] Hassan A-E., Richard C-H., *Architecture Recovery of Web Applications*. Proceedings of the 24th International Conference on Software Engineering. Pages: 349-359. 2002.

[10] Hirst G., St-Onge D., *Lexical chains as representations of context for the detection and correction of malapropisms*. In Fellbaum, C., ed., WordNet: An electronic lexical database. MIT Press. Pages: 305–332. 1997.

[11] Jiang, J., and Conrath, D., *Semantic similarity based on corpus statistics and lexical taxonomy*. In Proceedings on International Conference on Research in Computational Linguistics, Pages: 19-33. 1997.

[12] Leacock C., Chodorow M., Combining local *context and WordNet similarity for word sense identification*. In Fellbaum, C., ed., WordNet: An electronic lexical database. MIT Press. Pages: 265-283. 1998.

[13] Lin, D., *An information-theoretic definition of similarity*. In Proceedings of the International Conference on Machine Learning. 1998.

[14] Malki M., Ayache M., Rahmouni M.K., *Rétro-ingénierie des Bases de données relationnelles : Approche basée sur l’analyse de formulaires*. 17th Congrès INFORRSID, pp. 55-74, Toulon, La Garde June 1999, France.

[15] Malki M., Flory A., Rahmouni M.K., *Static and Dynamic Reverse Engineering of Relational Database Applications: A Form-Driven Methodology*. In Proc. Of Int. Conf. AICCSA’01, IEEE Computer Society Press, pp. 191-194, Beyrout 2001a, Liban.

[16] Malki M., Flory A., Rahmouni M.K., *A Form-Driven Reverse Engineering of Relational Databases*. In Proc. Of Int. Conf of Retis’01, OCG Oesterreichische Computer Gesellschaft, Austrian Computer Society Press. pp. 201-206, Lyon 2001b, France.

[17] MALKI M., *Rétro-ingénierie des applications – bases de données relationnelles : Approche dirigée par l’analyse de formulaires.* Thèse doctorat d’état 2002.

[18] Malki M., Flory A., Rahmouni M.K., *Extraction of Object-Oriented Schemas from Existing Relational Databases: A Form-Driven Approach*. INFORMATICA International Journal (Lithuanian Academy of Sciences), pp 47-72, Vol. 13(1), 2002.

[19] Marc E., Peter H., Mark H., Nenad S., *Similarity for Ontologies -a Comprehensive Framework*. In 13th European Conference on Information Systems. May 2005.

[20] Patwardhan S., Banerjee S., Pedersen T., *Using measures of semantic relatedness for word sense disambiguation*. In Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics. Pages: 241–257. 2003.

[21] Resnik P., *Using information content to evaluate semantic similarity in a taxonomy*. In Proceedings of the 14th International Joint Conference on Artificial Intelligence. Pages: 448–453. 1995.

[22] RUMBAUGH J. et al., *Modélisation et conception orientées objet (OMT)*. Edition française revue et augmentée en 1994.

[23] Wu Z., Palmer M., *Verb semantics and lexical selection*. In 32nd Annual Meeting of the Association for Computational Linguistics, Pages: 133–138. 1994.

[24] Giuliano Antoniol, Massimiliano Di Penta and Michele Zazzara.: *Understanding Web Applications through Dynamic Analysis*. Proceedings of the 12th IEEE International Workshop on Program Comprehension (IWPC'04). 2004.

[25] G.A. Di Lucca, A.R. Fasolino, P. Tramontana, U. De Carlini. *Abstracting Business Level UML Diagrams from Web Applications*. Proceedings of the 11th IEEE International Workshop on Program Comprehension (IWPC'03). 2003.