

A Framework for Exploiting Conceptual Modeling in the Evaluation of Web Application Quality

Pier Luca Lanzi, Maristella Matera, and Andrea Maurino

Dipartimento di Elettronica e Informazione, Politecnico di Milano
P. zza L. da Vinci 32, 20133 - Milano - Italy
{lanzi,matera,maurino}@elet.polimi.it

Abstract. This paper illustrates a method and a toolset for quality evaluation of Web applications that exploits conceptual specifications, deriving from the adoption of model-based development methods, for the evaluation in pre- and post-delivery phases.

Keywords: Conceptual Modeling, Web Application Quality, Web Usage Mining.

1 Introduction

The ever-increasing spread of the Web asks for new methods for improving the quality of Web applications. Conceptual modeling improves the quality of final applications by fostering regularity and the definition and reuse of effective solutions. However, little attention has been put on using conceptual specifications for enhancing the evaluation activities occurring throughout the whole development process.

We have defined a model-based framework that exploits conceptual schemas for evaluating Web applications both at design time and after the application deployment [6, 7]. This paper illustrates some recently introduced components that assist evaluation activities performed after application deployment. More specifically, such components elaborate Web usage logs enriched through meta-data deriving from the application conceptual schema.

Our evaluation framework has been defined for a specific conceptual model, WebML [2], and has been implemented extending a commercial CASE tool [8]. WebML offers a set of visual primitives for defining conceptual schemas that represent the organization of the application contents and of the hypertext interface. The organization of data is expressed through the Entity-Relationship model (E-R). The hypertext is then specified by composing elementary pieces of contents retrieved from the database, called *content units*, to form *pages*. WebML primitives are also provided with an XML representation that specifies additional properties, not conveniently expressible in the visual notation.

This paper introduces the Web log analysis covered by our evaluation framework, and shortly describes the architecture of an accompanying tool-set supporting the automatic execution of the proposed evaluation method. A deeper description, as well as results of applying the quality evaluation method to real-life Web applications can be found in [5].

2 The Evaluation Framework

Our evaluation framework supports three kinds of analysis, based on the knowledge of the application conceptual schema.

In the pre-delivery phase, the *Design Schema Analysis* (DSA) verifies the correctness and the internal coherence of specifications [3,6]. It operates directly on conceptual schemas by looking for design errors and inconsistency.

In the post-delivery phase, evaluation still exploits the schema knowledge. Thanks to an advanced logging mechanism extending the runtime engine of WebML applications, Web logs are enriched with meta-data related to the application conceptual schema, thus obtaining the so-called *conceptual logs* [7]. The evaluation then focuses on such enriched logs, according to two techniques:

- *Web Usage Analysis* (WUA) produces reports on content access and navigation paths followed by users.
- *Web Usage Mining* (WUM) applies mining techniques for discovering interesting (sometimes unexpected) associations between accessed data. The aim is to identify possible amendments for accommodating newly discovered user needs.

The peculiarity of the post-delivery evaluation is the exploitation of *conceptual logs*, defined as XML-based “enriched” Web logs that integrate the conventional HTTP log data, generated by Web servers in ECLF (Extended Common Log Format) format, and meta-data about the computation of page contents. As can be seen in Figure 1, for each requested page such meta-data include: *i*) identifiers of the page and of its content units, as resulting from the application conceptual schema, that provide references to detailed properties defined in the conceptual schema but not traced by the logging mechanism; *ii*) primary keys of database instances used at runtime to populate content units.

DSA has been illustrated in [3,6]. Therefore, in the next subsections we concentrate on the two analysis approaches operating over *conceptual logs*.

2.1 Web Usage Analysis

WUA analyzes conceptual logs for computing access reports on user content access and user navigation paths. The main objective is identifying the contents most requested by users and evaluating if the hypertext design accommodates such user needs.

WUA comes in two flavors: *Access Analysis*, and *Navigation Analysis*.

Access Analysis computes traffic statistics, with the aim of verifying if the communication goals the Web application has been designed for are supported by the hypertext interface. The model-based approach, which distinguishes between data modeling and hypertext modeling, allows performing:

- *Data Access Analysis*: it computes statistics for the access to data schema entities and their specific instances.
- *Hypertext Access Analysis*: it focuses on the usage of the hypertext elements (content units, pages, areas) publishing specific data elements.

```

1 <Logs>
2 <Request id_P="3178">
3   <Page SchemaRef="page33"/>
4   <LocalTime>
5     ... ..
6   </LocalTime>
7   <User>
8     ... ..
9   </User>
10  <PageUnits>
11    <Unit>
12      <Unit_Id SchemaRef="data_unit84"/>
13      <DataInstance>17</DataInstance>
14    </Unit>
15    <Unit>
16      <Unit_Id SchemaRef="index_unit9"/>
17      <DataInstance>10</DataInstance>
18      <DataInstance>5</DataInstance>
19      <DataInstance>9</DataInstance>
20    </Unit>
21  </PageUnits>
22 </Request>
23 ... ..
24 </Logs>

```

Fig. 1. Extract from the conceptual logs.

It therefore results that our Access Analysis extends the statistics normally offered by state-of-the-practice traffic analyzers, which mostly address page visits, and do not log database instances populating dynamic pages.

Navigation Analysis then verifies if the hypertext topology supports content accessibility. It reconstructs navigation paths adopted by users for reaching core application contents, with the aim of identifying if end users exploit navigation paths embodied within the designed hypertext, or else adopt alternative access mechanisms. The reconstruction of the user interaction results to be precise and detailed, as it exploits the conceptual schema of the application hypertext. Also, reconstructed user paths, as well as the identified problems, are represented on top of the visual specification of the conceptual schema; this facilitates the comparison between the designed hypertext and its actual use by users.

2.2 Web Usage Mining

WUM operates on conceptual logs, and applies XML mining techniques [1] for discovering interesting (sometimes unexpected) associations among visited hypertext elements and accessed data. The execution of mining tasks produces:

- *XML association rules* of the form $X \Rightarrow Y$, stating that when the log element X (called the *rule body*) is found, it is likely that the log element Y (called the *rule head*) will be also found. Depending on the adopted mining statement, the retrieved association can be related to database entities or instances, hypertext components (areas, pages, content units), or also hypertext components coupled with their populating data instances.

- *XML sequential patterns*, whose rule body and head are also bounded to their position in the log sequence, thus representing temporal relations.

Based on such rules, so far we have focused on two mining tasks:

- Finding data that are often accessed together, considering as transaction a user request, implemented through the mining of association rules between data entities and instances accessed within the same user session. It is worth noting that such associations are not easily discovered in traditional logs that do not record data instances used to populate dynamic Web pages, and generally require several post-processing efforts.
- Analyzing user navigation sequences for accessing some core information contents, by mining sequential patterns related to sequences of pages and content units within the same user session. The WebML characterization of information concepts and content units allows filtering sequences, concentrating the analysis on relevant navigation paths leading to some selected core concepts.

3 The Framework Architecture

The software architecture of our evaluation framework is organized in three distinct layers:

- The *Data Extraction Layer* gathers inputs needed for evaluation (Web server access, logged execution data, the application conceptual schema and the application data source), and transforms them into the format required by the three analysis techniques. It also generates conceptual logs.
- The *Analysis Layer* includes software components for the execution of DSA, WUA and WUM over data gathered through the Data Extraction Layer.
- The *Result Visualization Layer* allows evaluators to invoke the different analysis tasks and shows graphically the analysis results, through a graphical user interface.

Some XML repositories enable the interaction between layers:

- The *Analysis Data Warehouse* stores inputs gathered and elaborated by the Data Extraction Layer, represented in XML format.
- The *Result Warehouse* stores the results produced by the Analysis Layer in XML format. Such data are then used by the graphical user interface for generating and visualizing the analysis reports.
- The *Analysis Tasks Repository* stores the analysis procedures that can be expressed both in XSL and XQuery.

The ubiquitous use of XML technologies improves the number of strategies the evaluator can adopt in order to manipulate and query data. Also, the quality evaluation framework results to be very flexible and extensible: new analysis tasks can be easily specified and added to the framework. Therefore, each design team can define its own quality criteria, code their measures in XSL or XQuery, two extensively used W3C standards, and adding them within the the *Analysis Tasks* repository. Additionally, the use of warehouses between layers improves the framework extensibility, since it is possible to add new software modules, for example for performing new kinds of analysis, without affecting other components.

4 Conclusion

During last years, several methods and tools have been proposed for the analysis of Web logs [4]. The majority of them are however traffic analyzers. In addition to calculating traffic statistics, our Web Usage Analysis is able to compute advanced statistics, related to database entities and instances, and to hypertext components of any granularity.

Thanks to the intensive use of the application conceptual schema, our framework introduces a number of advantages also with respect to Web usage mining. Several data mining projects have demonstrated the usefulness of a representation of the structure and content organization of a Web application [4].

Our future work will concentrate on the incremental enrichment of the statistics and mining tasks for analyzing Web usage data. We are also working on the improvement of the graphical user interface, for allowing designers to define new analysis tasks though a visual paradigm, without the need of manual XSL and XQuery programming.

References

1. D. Braga, A. Campi, S. Ceri, M. Klemettinen, and P. Lanzi. A Tool for Extracting XML Association Rules. In *Proc. of ICTAI'02, November'02, Crystal City, USA*. IEEE Computer Society, 2002.
2. S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, and M. Matera. *Designing Data-Intensive Web Applications*. Morgan Kauffmann, 2002.
3. S. Comai, M. Matera, and A. Maurino. A Model and an XSL Framework for Analyzing the Quality of WebML Conceptual Schemas. In *Proc. of the ER'02-IWCMQ'02 Workshop, Tampere, Finland, October'02*, LNCS. Springer, 2002.
4. R. Cooley. The Use of Web Structures and Content to Identify Subjectively Interesting Web Usage Patterns. *ACM TOIT*, 3(2), May 2003.
5. P. Fraternali, P. Lanzi, M. Matera, and A. Maurino. Model-Driven Web Usage Analysis for the Evaluation of Web Application Quality. Technical Report, Polytechnic of Milan, April 2004.
6. P. Fraternali, M. Matera, and A. Maurino. WQA: an XSL Framework for Analyzing the Quality of Web Applications. In *Proc. of ECOOP'02-IWWOST'02 Workshop, Malaga, Spain, June '02*, 2002.
7. P. Fraternali, M. Matera, and A. Maurino. Conceptual-level Log Analysis for the Evaluation of Web Application Quality. In *Proc. of LA-Web'03, Santiago, Chile, November'03*. IEEE Computer Society, 2003.
8. WebRatio. <http://www.webratio.com>.