A Quality Aware Service-oriented Web Warehouse Platform

Adriana Marotta
Universidad de la Republica
J. Herrera y Reissig 565
Montevideo, Uruguay
(598) 27114245
amarotta@fing.edu.uy

Laura González
Universidad de la Republica
J. Herrera y Reissig 565
Montevideo, Uruguay
(598) 27114245
lauragon@fing.edu.uy

Raúl Ruggia
Universidad de la Republica
J. Herrera y Reissig 565
Montevideo, Uruguay
(598) 27114245
ruggia@fing.edu.uy

ABSTRACT
In order to be a useful tool, a Web Warehouse (WW) should take into account the quality of the data it manages and the quality of the services that provide the source data. It should also have enough flexibility to endure the high volatility of web sources. In this work we propose a WW platform that satisfies these two conditions. It manages data and services quality, measuring quality at the different stages of the system life-cycle and at the same time, using these measures as input for information processing. Additionally, the platform achieves flexibility and configurability because it is strongly based on information-processing services.

Categories and Subject Descriptors
H.0 [Information Systems]: General

General Terms
Design

Keywords
Web Warehouse, Data Services, Data Warehouse, Data Quality, Service Quality.

1. INTRODUCTION
Supported by new technology trends, like Web 2.0, Cloud Computing and Web Services, information available on the Web is increasing every day. Consequently, Web Warehouses (WW), Data Warehouses (DW) which consolidate data from the Web [7,11], have become a valuable tool for decision making in many areas. However, given the dynamic and autonomous nature of Web Data Sources (WDSs) [2], the process of building a WW presents major challenges. First, web data can be delivered through many heterogeneous formats and protocols, including among others HTML pages, XML documents, RSS or ATOM feeds, Web Services, raw data, LinkedData, etc. Second, quality of published data is often inadequate, since there exists a wide variety of problems such as data typing errors, missing values, obsolete and inconsistent data. Also, autonomy of WDSs usually leads to uncertainty in terms of availability, cost and unexpected changes in the format or protocols they use to deliver data.

In order to be a useful tool, a WW should take into account the quality of the data it manages and the quality of the services that provide the source data. It should also have enough flexibility to endure the high volatility of web sources, which may force the system to change its sources or to change the process it applies to the data obtained from certain source. In this work we propose a WW platform that satisfies these two conditions.

The proposed platform manages data and services quality, measuring quality at the different stages of the system life-cycle and at the same time, using these measures as input for information processing. In particular, data and services quality measures are used for sources selection, data integration, and for providing data quality information to the WW data consumers.

In order to achieve flexibility and configurability the proposed platform is designed with a service-oriented approach. The different tasks that are performed for processing data that will feed the DW and later the multidimensional cubes, are implemented as data and information specialized services. These services are dynamically selected by the system according to the information processing needs, which can vary in each WW loading.

The main contributions of this paper are the characterization of the life-cycle of the WW, the definition of a quality management framework, and the specification of a service oriented platform aiming to provide an implementation basis.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 focuses on WW life-cycle and the reference architecture. Section 4 explains the quality management framework. Section 5 depicts the service oriented platform, and Section 6 concludes.

2. RELATED WORK
Various WW platforms have been proposed in the literature dealing with specific problems of WW development. For example, in [3] and [2] the authors describe the project Whoweda (Warehouse of Web Data) whose objective is to design and implement a WW that materializes and manages useful information from the Web. Besides, in [13] the author presents a framework for warehousing selected Web content, using a partially materialized approach and an extended ontology to achieve data integration. Finally, in [11] the authors propose a wrapper - mediator architecture in order to minimize the impact of
Web sources changes on the DW schema. None of these works consider quality as an integral issue during the whole life cycle of a WW, neither propose an architecture based on data and information services and their interaction. The service-oriented architecture proposed in this paper considers quality management aspects starting from the web data sources and all through the different stages until the end user interacts with the WW.

Data quality in Web sites has been analyzed in different works [6, 14]. Quality of Service aspects are analyzed and identified in [9, 5]. In contrast to these works, ours focuses in a WW context and considers quality factors regarding various types of web data sources and identifying which of them are pertinent in a WW scenario. We also analyze for which stages, during the WW life cycle, these quality factors are relevant, determining which of them might be propagated to the WW to provide value added information to the WW final user.

3. WW LIFE-CYCLE

Due to the complexity of this kind of systems it is important to specify the different stages that compose its life-cycle, as well as the different roles that actively participate in its installation and operation. In this section we first explain the WW architecture and then the life-cycle of the system.

The WW we propose is based on the architecture proposed in [10], shown in Figure 1, which address the three main tasks of the system: i) data extraction, (ii) data integration and (iii) data transformation and loading to the DW, through different modules.

The Data Service Infrastructure (DSI) module provides the execution environment to implement, host and execute Data Services (DSs), which access and extract data from the web data sources (WDSs). The DSI module has three main responsibilities. First, it has to provide the mechanisms to extract data from different types of WDS and expose them as homogeneous DSs using standard technologies. Second, the DSI module has the responsibility to monitor the DSs and periodically measure quality to allow a quality driven discovery of them. The measurement of data and services quality is performed by specific services that are invoked by the QM module. This involved quality information is stored as DQ and QoS metadata. Finally, the DSI provides runtime adaptation mechanisms, in the extraction process, to react to possible degradation in DSs’ quality.

The Integrator module, which invokes the DSs, carries out data integration and provides data in a normalized format, generating the Integrated Database (IDB). Its main responsibilities are quality-oriented selection of DSs, data integration, and generation of quality metadata associated to the integrated data. The Integrator interacts with expert users, who perform the main data integration decisions, such as object identification and conflict resolution. This interaction is achieved through a rich user interface.

The ETL and OLAP modules carry out the classic functions of DW context. In addition they have to perform quality metadata propagation. In order to do this, they have to capture QoS and DQ metadata associated to the processed data. In turn, DWQM module achieves data quality measurement, receiving data quality metadata as input and combining it with new measurements performed over the DW through the invocation of specific DW quality measurement services.

We propose a platform that gives support for all the tasks needed for achieving the WW system. It is a general platform that must be instantiated for the different domains for which the WW will operate. This instantiation consists, for example, on the definition of the DW schema and the schema that will be fed by the DSI data services with data extracted from the web.

Therefore, once the WW platform is implemented, some tasks must be accomplished before it can be put in operation. First, a preliminary set of web sites of the domain of interest must be selected. Then, the platform must be installed and configured in order to be instantiated for the particular domain. After that, the WW loading may be executed; this includes web data extraction, integration, transformation and loading into the DW. Finally, the WW can be exploited by the final user. All these stages compose the life-cycle of the WW.

Figure 2 shows the WW life-cycle stages with the different roles participation. We define the following four stages:

1) Definition of a set of domain-specific web data sources. In this stage a first set of web data sources is selected, taking into account quality characteristics, such as cost and reputation. We call them the candidate WDSs. We further explain this step in Section 5.
2) System installation and configuration. In this stage the WW platform is instantiated for a particular domain and requirements. This involves the following main tasks: (i) DW design and definition (this includes schema and ETL processes), (ii) design of the schema that will be fed by the DSI DSs, (iii) DSs development, and (iv) DSs selection. In (iii), DSs are developed for the candidate WDSs. In (iv), DSs that will populate de WW are selected, taking into account quality characteristics that are deduced from the successive execution of them, such as performance or content accessibility. We further explain these two steps in Section 5.

3) Extraction, Integration, Transformation and Loading with quality management. This stage is when the data flows from the Web, throughout the system towards the DW. This execution may occur periodically or it may occur each time a data update occurs in the WDSs. As shown in Figure 2, from this stage it is also possible to return to stage 1, if one wants to change the candidate WDSs, for example adding new web sites as sources. During this processing, quality metadata is considered for DSs invocation, for integration tasks, and besides it is enriched and propagated to the DW.

4) WW exploitation. The WW is exploited, through OLAP tools that allow navigating data and applying multidimensional operations over data, for analysis. The built data cubes may also be published on the web, being available to any user.

During the described life-cycle, different kinds of users interact with the system. The IT Professional must participate in stages 1 and 2, being necessary in all the tasks that are previous to the loading and exploitation of the WW. The Domain Expert must help with his knowledge when candidate WDSs are selected (stage 2 of the life-cycle), as well as at the moment of DSs selection and information integration (stage 3 of the life-cycle). During data integration, this user will help with conflict resolution and object identification problems. Lastly, the Final User will take profit of the constructed WW.

4. QUALITY MANAGEMENT

After reviewing the literature, we notice that Web data quality factors can be organized in two broad categories: service related factors and data related factors. The service related factors considered in this work are selected from quality related literature regarding Web Services [5, 9], Web Applications [6] and Mashups [4]. We organize these factors in six quality dimensions: Service Level (S.L.), Interoperability (I.), Security (S.), Business Value (B.V.), Usability (U.) and Stability (St.). In turn, the data related factors considered within this work is the result of a deep analysis of data quality dimensions literature [1, 5, 12, 14], combined with our own experience in data quality models definition. We organize data quality factors in six quality dimensions: Accuracy, Completeness, Freshness, Uniqueness, Consistency and Reliability. The complete list of factors along with their definitions is presented in [10].

Throughout the selection and extraction process, not all quality factors might be relevant for a specific task. We analyze whether the identified quality factors are relevant for the following tasks:

- Selecting Candidate WDS. The relevant factors are the ones used to decide if it is worth building a wrapper for a WDS.
- Selecting DSs to Populate the WW. The relevant factors are the ones which are used by the specialized user to select the DSs to use.
- Propagating Quality Values to the WW. The relevant factors are the ones useful for WW final users.

It is important to note that this analysis considers a very general scenario. It is expected that the relevance of a quality factor might vary according to a specific context or domain.

As shown in Table 1, we found that all the identified service related factors are relevant for the WDS candidate selection. For the DS Selection, the Interoperability and Usability dimensions are not important anymore given that wrappers have already been developed. However, the rest of the factors remain relevant. Finally, we consider that the identified security factors (e.g. integrity) are important to propagate to the WW.

On the other side, we found that all data related quality factors are relevant for the three tasks. First, these quality data can be useful for deciding whether or not to develop a wrapper and it can also provide valuable information for selecting DSs. Additionally, the availability of this information can be an important resource for the final user.

<table>
<thead>
<tr>
<th>Dim.</th>
<th>Factor</th>
<th>WDS C. Selection</th>
<th>DS Selection</th>
<th>WW Propag.</th>
</tr>
</thead>
<tbody>
<tr>
<td>S.L.</td>
<td>All.</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>I.</td>
<td>All.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.</td>
<td>All.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>B.V.</td>
<td>Cost</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>U.</td>
<td>All.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>St.</td>
<td>All.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. SERVICE ORIENTED PLATFORM

The proposed architecture leverages the service-oriented approach to perform its main tasks.

5.1 Web Data Selection and Extraction

The selection and extraction process is mainly supported by the DSI module, which provides an execution environment to host DSs and mechanisms that aid in describing, publishing, discovering and invoking DSs. The DSI module manages DSs quality to support a quality aware discovery of DSs, and provides a runtime adaptation mechanism aiming to maintain the DSs quality level. Furthermore, it provides quality information that can be propagated to the WW. This quality driven process is composed of four main steps.
The first step consists in selecting Candidate WDSs. This selection is performed taking into account the quality factors presented in Table 1. The second step consists in the Development and Deployment of Data Services. In order to provide a homogeneous access to various types of WDSs, we propose implementing Data Services (DSs) to encapsulate the extraction logic that wrappers might implement. To support the development of DSs, the DSI module offers an execution environment which allows to measure DS quality and eventually to take corrective actions in case of DSs quality degradation.

The third step is the Data Services Selection, which is aided by two DSI components: the Data Service Registry (DSR) and the Data Service Quality Management (QM) component. The DSR component provides functionalities to search and discover the deployed DSs. The QM component continuously monitors DSs quality. The quality information computed by the QM module is then made available to the DSR component and the Integrator module, so it can be propagated to the WW.

The last step is the Data Services Invocation. Given that quality information can change over time, the DSI module supports a quality driven adaptation mechanism aiming to maintain a certain quality level.

5.2 Quality Measurement
In the proposed platform, quality measurement is carried out through the invocation of specialized services. We follow the approach proposed in [8], where the functionalities of quality tools are described as abstract quality services and a delegation mechanism binds an abstract quality service to a specific implementation in a given external quality tool. With this mechanism, a quality service is an implementation of a quality functionality that can be either custom implemented within an organization or provided by external quality tools, such as QM and DWQM (see Figure 1), which are in charge of quality measurement, at the moment of data extraction and after data is loaded in the DW, respectively.

6. CONCLUSION
Although WW have been largely studied, issues like data and service quality implications and data life-cycle modeling have been much less addressed.

The proposed quality management approach takes into account data and service quality providing key information about the practical usability of external sources. In addition, quality information is used to filter undesirable items and also to guide the processes of source selection and extraction. Our approach is based on Service Oriented Architectures and on new technologies like Web Services, XML and middleware platforms, which facilitate accessing data sources distributed in the Web as well as processing the extracted data.

This paper intends to be a step forward on developing a novel approach to data quality aware Web Warehouses leveraging Service Oriented Architectures and associated technologies.

7. REFERENCES