Web warehouse – a new web information fusion tool for web mining

Lean Yu a,b,*, Wei Huang c, Shouyang Wang a,d, Kin Keung Lai b,d

a Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 55, Zhongguancun East road, Beijing 100080, China
b Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
c School of Management, Huazhong University of Science and Technology, Wuhan 430074, China
d College of Business Administration, Hunan University, Changsha 410082, China

Received 20 June 2005; received in revised form 8 October 2006; accepted 24 October 2006

Abstract

In this study, we introduce a web information fusion tool – web warehouse, which is suitable for web mining and knowledge discovery. To formulate a web warehouse, a four-layer web warehouse architecture for decision support is firstly proposed. According to the layered web warehouse framework architecture, an extraction–fusion–mapping–loading (EFML) process model for web warehouse construction is then constructed. In the web warehouse process model, a series of web services including wrapper service, mediation service, ontology service and mapping service are used. Particularly, two kinds of mediators are introduced to fuse the heterogeneous web information. Finally, a simple case study is presented to illustrate the construction process of web warehouse.

Keywords: Web warehouse; Web mining; Extraction–fusion–mapping–loading process; Wrapper service; Mediation service; Mapping service; Semi-structured data

1. Introduction

In recent years, we have witnessed an immense growth in the availability of on-line information at an unprecedented pace, which makes the World Wide Web (WWW) become a vast information repository about all areas of interest. In this information store, only a few parts are structured data, which are mainly from the organizational online transaction processing (OLTP) system. Basically, these structural data are extracted from operational systems, transformed, and loaded into the data warehouse or data mart according to a specified schema, such as the star or snowflake schema [1]. The data warehouse [1] can be used for business intelligence, such as business decision-makings, based upon knowledge created by data mining and knowledge discovery technologies.

Besides structured data, a large percent of web information on the Internet is semi-structured textual information, and they are stored as static hypertext markup language (HTML) pages that can be viewed through a browser. Although some websites provide search engines, their query functions are often limited, and the results return as HTML pages [2]. Even though there are some extensible markup language (XML) [3] pages in the collected web information, we can also convert XML into HTML using an online conversion tool provided by Serprest [4]. However, the information is not neglected by business organizations because they contain much hidden valuable knowledge which can be used for business decision-makings. Furthermore, a recent investigation conducted by Delphi Group [5], showed that 80% of a company's information is represented in a textual format and many organizations have been utilizing the web information from the Internet for decision-making. As more and more business organizations move...
their operations to the Internet, these semi-structured web information or HTML pages that they generated will play an increasingly important role in providing the enterprise managers with up-to-date and comprehensive information about their business domain. In a sense, the information collected from the web must be incorporated into the data warehouse (which in this case will be more properly called web warehouse [6]) for business decision-makings [7]. Furthermore, in order to support high-level decision-making, users need to comprehensively utilize and analyze the data from various sources [8]. This also motivates the creation of web warehouses, which are data warehouses that contain consolidated data obtained from web sources [9,10].

Designing a data/web warehouse entails transforming the schema that describes the source data into a multidimensional schema for modeling the information that will be analyzed and queried by users [11]. Actually, the construction of a web warehouse is a process of web information fusion that integrating web information from different sources into web warehouse. Thus, a web warehouse used in this paper refers to a single, subject-oriented, integrated, time-variant collection of web information that supports the web mining and knowledge discovery. Due to the fact that web information does not only include structural data but also semi-structured text, web warehouse can be seen as a federated warehouse integrating data warehouse [1,12,13] and text warehouse [14]. In this study we propose a generic framework architecture to design a web warehouse, similar to the construction process of data warehouse [1]. In the proposed framework for web warehousing, we view the Internet or the WWW as data sources. The information including structured data and semi-structured text is extracted from the web, refined, transformed, and placed into the web warehouse for later use. In our proposed framework, a configurable extraction program presented by [2] is first used for converting a set of hyperlinked HTML pages (either static or results of queries) into database objects. After extraction, the converted results are refined, integrated and transformed to appropriate forms or formats and then loaded into the web warehouse for web mining and knowledge discovery purposes.

Although web warehouse is new relative to data warehouse, some approaches concerning related issues have been proposed in the literature. Bhowmick et al. [6] utilized a database approach to discussing some design issues of web warehouse. Lim et al. [15] used a data warehousing approach to handling web information. In [11], Vrdoljak et al. proposed a semi-automated methodology to design web warehouses from XML schema. Similarly, Golfarelli et al. [16] outlined a conceptual design technique starting from document type definition (DTD) [17] in XML sources. In [18], Bhowmick gave a data model and algebra architecture for a web warehouse in his PhD thesis. In addition, some web information extraction methods such as configurable extraction program [2] and mediation technology [19], and some web information integration or fusion approaches, such as paraphrasing [20] and ontology-based technology [21] can also be used as some important tools for web warehousing. However, these models and methods only focus on a specified problem of web warehouse. For example, Bhowmick et al. [6] only discussed several design issues about web warehouse. Hammer et al. [2] and Victorino [19] only focused on information extraction problem while Barzilay [20] and Wache et al. [21] only presented the information integration problem in the web environment. Although some studies (e.g., [8,13]) pointed out that a web warehouse can be built based on XML format, the HTML format still dominates the web world. The distinct evidence is that large volumes of web data are presented in HTML format rather than XML even if XML will become a standard for the exchange of semi-structured data in the near future.

Different from previous studies in the literature, our study tries to create a HTML-based approach for web warehouse construction. In order to integrate web information into the federated data warehouse, i.e., web warehouse, an extraction-fusion-mapping-loading (EFML) process model for web warehouse is proposed. The main aim of this study is to provide assistance to accomplish those processes or tasks, and more specifically to address the problems of constructing web warehouses. Based on this motivation, we present a general framework architecture of web warehouse for decision support that covers all steps starting with information extraction from the web and finishing with the web warehouse. Note that the web information extraction mechanism gathers HTML pages about the user-specified domain from the web and generates mappings from web sources to an integrated schema. We utilize existing techniques developed in the database area, such as those proposed in [2,22] for extracting information from web pages [23]. Finally, a specialized component performs the mapping from the integrated source schema to the web warehouse schema [24], based on existing data warehouse design techniques [1,12,13,25].

The main contribution of this paper is to propose an EFML process model to perform web information extraction, integration, mapping and loading and thus formulate a web warehouse for web mining and knowledge discovery. Because the ultimate goal of the web warehouse is to support decision-making for users, we organize the rest of the paper as follows. Section 2 presents a generic web warehousing formulation process for web mining and decision support. In Section 3, we illustrate the proposed web warehousing process through a case study. Section 4 concludes the article.

2. The generic formulation process for web warehousing

2.1. The general web warehouse architecture for decision support

The emergence of web warehouse architecture is to respond the evolving data and web information requirements. Initially, the classic data warehouse was used to...
extract transactional data from operational systems to perform on-line analytical processing (OLAP) [7,12]. Because there are different data types, such as structured data and semi-structured text, among web information, the traditional data warehouse, which only handles the structured data, does not deal with semi-structured texts. Therefore, a new warehouse architecture should be presented to meet the practical requirements. In such situations, web warehouse is proposed in response to the increasing web information. For this point, data warehouse will be evolved into a federated data warehouse [7], i.e., web warehouse. Typically, the general web warehouse architecture for decision support is shown in Fig. 1.

As can be seen from Fig. 1, the general web warehouse architecture for decision support consists of four layers: data source layer, warehouse construction layer, web mining layer and knowledge utilization layer. The data sources layer of web warehouse is composed of the organization’s internal data sources including daily operation data, internal files and OLTP data, etc. and external web text repositories and electronic messages. This layer mainly provides a data foundation for web warehouse construction. Note that a distinguishing and relevant issue of web warehouses is that their data sources are mainly external, while sources in classic data warehouses are mainly internal to the organization. It is therefore, very important to maintain an integrated schema, which represents a unified view of the data. As a direct consequence, information fusion follows the local as view approach, where each data source is defined as a view of a global integrated schema.

In order to formulate a global view for internal and external data, the information fusion will be inevitable. But web warehouse construction must extract some related information firstly, and then integrate some conflicted data and subsequently transform or map the integrated information into the appropriate schemas. Finally, the transformed schemas are loaded into warehouse to construct the web warehouse. That is, the web warehouse is constructed by using an extraction–fusion–mapping–loading (EFML) process model based upon a series of services, which will be further addressed in the subsequent subsection. In the EFML process of web warehouse construction, the fusion process is to integrate the heterogeneous web information using mediation service [19] or ontology service [21]. In this sense, the proposed web warehouse can be used as an alternative web information fusion tool.

When the information extracted from the web have been loaded into the web warehouse through an EFML process, some mining methods, such as classification, clustering, association analysis, and OLAP can be used to explore the hidden knowledge and thus formulating a knowledge repository. As earlier noted, the ultimate aim of web warehouse is to support decision-making for users. In the last layer, users can search and query knowledge repository via knowledge portals and obtain the corresponding decision information to help decision-making. In the following

---

**Fig. 1.** The general web warehouse architecture for decision support.
subsection, we focus on the web warehouse construction with EFML process model.

2.2. EFML process model for web warehouse construction

Based upon the general framework architecture from Fig. 1 proposed in the above subsection, we can find that the web warehouse can actually be built by an extraction–fusion–mapping–loading (EFML) process model relying on a series of services. Generally, the overview of EFML process model for web warehouse construction is presented in Fig. 2. It is worth noting that Fig. 2 is only an unfolded illustration of Layer II in Fig. 1.

The EFML process model associated with web warehousing consists of five well-defined activities: (1) exacting the related data from the web pages, (2) integrating web information from various sources and types, (3) transforming or mapping the information into the appropriate schema, (4) assisting in the refinement of data and information schema, and (5) loading the refined schema into web data warehouse. These activities are illustrated in detail in the following.

2.2.1. Extraction

In the information extraction, wrapper service plays a key role in extracting the web information. The goal of a wrapper is to access a source, extract the relevant data and present such data in a specified format. In this study, a configurable extraction program proposed by [2] for converting a set of web pages into database objects is used as a extractor or wrapper to retrieve the relevant data in object exchange model (OEM) [26] format, which is particularly well suited for representing semi-structured data. Of course, other web information extraction tools such as YACC [27] and Python [28] can also provide the wrapper services.

This wrapper takes a specification as an input that declaratively states where the data of interest is located on the HTML pages, and how the data should be “packaged” into objects. This wrapper is based on text patterns that identify the beginning and end of relevant data. Due to the fact that the extractor does not use “artificial intelligence” to understand the web contents as [2] indicated, the extractor can be used to analyze large volumes of web information. In the proposed wrapper service, the configurable extraction program as an extractor parses the HTML page based on the specification file. Fig. 3 shows a basic structure of the specification file. Note that the line numbers shown on the left-hand side of the figure are not part of the content but have been added for convenience of following discussions.

From Fig. 3, we can see that the attractor specification file consists of a sequence of commands, each defining one extraction step. Each command is represented by [variables, source, pattern], where source specifies the input text to be considered, pattern tells us how to find the text of interest within the source, and variables are one or more extractor variables that will hold the extracted results. The text in variables can be used as the inputs for subsequent commands. Taking Fig. 3 as an example, the extractor specification file consists of two commands delimited by bracket, the first command (lines 1–4) fetches the contents of the source file whose URL is given in line 2 into the variable called variable_1. The character “#” in line 3 means that everything (i.e., all contents of the HTML file) is to be extracted and saved. After the file has been fetched and its contents are read into variable_1, the extractor will...
filter out unwanted data such as the HTML tags and extra uninteresting text. The second command (lines 5-8) specifies that the result of applying the pattern in line 7 to the source variable variable_1 is to be stored in a new variable called variable_2. The pattern can be illustrated as “discarding everything until the first occurrence of a specific HTML tag and saving the information that is stored between the two HTML tags”. Here the character “*” represents that the information before a specified HTML tag should be discarded. Note that we can use multiple HTML tags to navigate the correct position that information begins to be stored.

Fig. 3 only lists a simple example of the specification file format; more commands can be added to extract the related information correctly from web pages. After the last command of the specification file is executed, some subsets of the variables with identical data structure will hold the data of interest. Further details about the extraction process will be illustrated in the case study.

2.2.2. Fusion

When the data are extracted through the wrapper services, many structured data with different schemas can be obtained. In order to formulate a unified view for these extracted data from various sources, it is very important to maintain an integrated schema. In this study, we use two kinds of mediation services to integrate web information from different sources. The two kinds of mediation services are designed to fuse schemas without and with structural heterogeneities. The former ones, called as “m”, are mediation services that fuse data with similar structures or similar schemas. The latter ones, called as “M”, are typical mediation services that integrate heterogeneous structure or heterogeneous schema information. In the situation of dynamic evolution, the m mediators can integrate the results returned by each wrapper and while the M mediator can fuse the schemas and the data returned by the m mediators. Through these two kinds of mediators, an integrated schema that reconciles structural heterogeneous information can be generated. A direct graphical illustration can be shown in Fig. 4.

In order to solve data conflicts in information fusion we follow the approach presented in [29], which is based upon a conceptual representation of the data warehouse application domain. The main idea is to declaratively specify suitable matching and reconciliation operations to be used in order to solve possible conflicts among data in different sources.

Another solution to information fusion is ontology-based services [21]. The goal of using ontology services is to resolve mainly the heterogeneity problem by performing mediation processes. These processes exploit some formal ontologies which take important part in the architecture. Ontology services aims to define the semantic description of services using ontological concepts. According to Gruber [30], an ontology is defined as an explicit and formal specification of a conceptualization. In general, the construction of domain-specific ontology is of utmost importance to providing consistent and reliable terminology across the warehouses. Similarly, hierarchical taxonomies are an important classification tool and here it is used as information integration tool. It can assist analysts in identifying similar, broader or narrower terms related to a particular terms thereby increasing the likelihood of fusing similar information from different information sources.

In addition, active rules and heuristics associated with object types as well as their attributes and functions can also be used for information integration. Content fusion in heterogeneous environments can make use of ontologies, particularly in the area of catalog integration [31,32].

2.2.3. Mapping

The mapping or transformation process is an important procedure in constructing web warehouse. In this study, the

![Fig. 4. Information fusion with different schemas.](image)
data warehouse schema is designed by applying a set of high-level transformation schema to the integrated schema. These transformation schemas embed meaningful data warehouse design techniques [1,25]. The whole data warehouse design framework is presented in [24]. Then, the designer applies these transformations or mapping services in terms of the desired design criteria, such as normalization design and de-normalization design. In a normalization design, one fact is stored in one place in the system. The advantages of the normalization design are that it can avoid redundancy and inconsistency. In a de-normalization design, one fact may be stored many place in a system. A de-normalized design has much redundant data. Usually a de-normalized relational design is preferred when browsing data and producing reports [1]. According to this transformation or mapping services, database tables are classified based on a dimensional model, e.g., a relation may be a dimension relation or a measure relation. An illustrative example about the mapping services will be presented in the next section. This mapping service facilitates the construction of structures that are specifically designed to satisfy warehouse’s requirements, and also provides design traceability.

Traceability is a quality measure that enables to improve design process as well as web warehouse management. For the design process, the trace behaves as a valuable documentation of the design and it can be very useful for design process reuse. For web warehouse management, the trace is a valuable tool for obtaining the mapping between the sources and web warehouse schema elements. This mapping is necessary at least for solving the following three problems in web warehouse management: (1) error detecting, (2) source schema evolution, and (3) data loading processes [8].

2.2.4. Metadata and loading

After extraction, fusion and transformation, the information is classified and indexed, and metadata is created in terms of domain concepts, relationships and events. In web warehouse, the domain contexts and domain usage constraints are specified. Data pedigree information is also added to the metadata descriptors, for example, intellectual property rights, data quality and source reliability etc. In addition, web mining and data analysis techniques can be applied to discover patterns in the data, to detect outliers and to evolve the metadata associated with object descriptors [7].

Usually, the metadata of the web warehouse consists of five kinds of information: (1) web pages classification criteria, (2) web warehouse design criteria, (3) mappings between the web pages and integrated global schema [33] and between the integrated schema and the web data warehouse [24], (4) the semantic and structural correspondences that hold among the entities in different sources schemas which is used for information integration or fusion[33,34], and (5) the classification of the web data warehouse schema structures according to the dimensional model [1,24]. In order to explore the hidden knowledge, the transformed or refined data, metadata, and knowledge should be loaded into web warehouse and stored in the web warehouse. For fast retrieval purpose, the loaded information should be indexed using multiple criteria, for example, by concept, by keyword, by author, by event type or by location. In the case where multiple users are supported, these should be indexed by thread, and additional summary knowledge may be added and annotated.

So far, a whole EFML process for web warehouse construction is completed. To give a direct view and understanding to this process model, an illustrative example is presented in the next section.

3. Case study – an illustrative example

3.1. Backgrounds

In this illustrative example, suppose that we need to make a decision about crude oil trading in terms of web quotation and trading volume information. To collect related information, we use a web site called New York Mercantile Exchange (NYMEX) (http://www.nymex.com/) as one of its information sources. This site reports the future price in the next few months and some historical trading volumes about crude oil future contracts. We focus on the light sweet crude oil futures in this study. As an example, a snapshot of the light sweet crude oil future quotation is shown in Fig. 5. Because the future quotation and historical trading volumes are not integrated in the same web page, we decide to use web warehouse to integrate these scattered information for decision support.

3.2. The EFML process for web warehousing

Since the quotation information is displayed by HTML format and it cannot be queried directly by users. Therefore, we first have to extract the contents of the quotation table from the underlying HTML page which is illustrated in Fig. 6. Note that the line numbers shown on the left-hand side of the figure and the next figure are part of the content but have been added to simplify the following descriptions and discussions.

Using the extractor based on the configurable extraction program [2] mentioned in Section 2.2, we can extract the quotation information from the web. Typically, the extraction process in this case is performed by five commands, as illustrated in Fig. 7. The first command (lines 1–4) is to fetch the contents of the source file whose unified resource location (URL) is given in line 2 into the variable called root. After the file has been fetched and its contents are read into root, the extractor will filter out unwanted data such as the HTML tags and extra uninteresting text. It is worth noting that the URL address “http://www.nymex.com/lsco_fut_cso.aspx” is the URL of Fig. 6 rather than the URL of time tags.
The second command (lines 5–8) specifies that the result of applying the pattern in line 7 to the source variable `root` is to be stored in a new variable called `quotation`. By performing the pattern definition in line 7, the variable `quotation` contains the information that is stored in line 18 and higher in the source file in Fig. 6 (up to but not including the subsequent `/table` token which indicates the end of the quotation table).

The third command (lines 9–12) instructs the extractor to split the contents of the variable `quotation` into “chunks” of text, using the string `tr align = left` (lines 7, 17, 27 etc. in Fig. 6) as the chunk delimiter. Note each text chunk represents one row in the quotation table. The split results are stored in a temporary variable called `_lscoquotation`. The underscore at the beginning of the name `_lscoquotation` indicates that this is a temporary table; its contents will not be included in the resulting OEM object. The split operator can only be applied if the input is composed of equally structured pieces with a clearly defined delimited separating the individual pieces [2].

![Fig. 5. A snapshot of the light sweet crude oil future quotation.](image)

```
1 <html>
2 <head>
3 <title>NYMEX: Light Sweet Crude Oil Quotation</title>
4 </head>
5 <body>
6 <table border="1" cellspacing="0" width="100%">
7   <tr align = left>
8     <td>April</td>
9     <td>May</td>
10    <td>June</td>
11    <td>July</td>
12    <td>Aug</td>
13    <td>Sep</td>
14   </tr>
15   <tr align = left>
16     <td>Last</td>
17     <td>Open High</td>
18     <td>Open Low</td>
19     <td>High</td>
20     <td>Low</td>
21     <td>Most Recent Settle</td>
22     <td>Change</td>
23   </tr>
24   <tr align = left>
25     <td>60.95</td>
26     <td>62.10</td>
27     <td>61.90</td>
28     <td>62.10</td>
29     <td>61.90</td>
30     <td>62.10</td>
31     <td>63.15</td>
32     <td>63.15</td>
33     <td>64.81</td>
34     <td>64.81</td>
35     <td>62.35</td>
36     <td>-1.79</td>
37   </tr>
38 </table>
```

![Fig. 6. The HTML source file of the crude oil future quotation.](image)
In the fourth command (lines 13–16), the extractor copies the contents of each cell of the temporary array into the array `lsco_quotation` starting with the second cell from the beginning. The first integer in the instruction `_lscoquotation` [1:0] indicates the beginning of the copying (since the array index starts at 0, 1 refer to the second cell), the second integer indicates the last cell to be included (counting from the end of the array). As a result, we have excluded the first row of the table which contains the individual column headings. Note that we can also filter out the unwanted row in the second command by specifying an additional * < html tag> condition before the “#” in line 7 of Fig. 7. The final command (lines 17–20) extracts the individual values from each cell in the `lsco_quotation` array and assigns them into the variables listed in line 17 (`time, t_url, last, open high, open low, high, low, most recent settle, changes`).

After the five commands have been executed, the variables hold the data of interest. This data is packaged into an OEM object, as illustrated in Fig. 8, with a structure that follows the extraction process. OEM is a schema-less model that is particularly well-suited for accommodating the semi-structured data commonly found on the web. Data represented in OEM constitutes a graph, with a unique root object at the top and zero or more nested subobjects. Each OEM object contains a label, a type and a value. The label describes the meaning of the value that is stored in this component. The value stored in an OEM object can be atomic or can be a set of OEM subobjects [2]. Interested readers can refer to [26] for more information about OEM.

Fig. 7. A specification file for extracting crude oil quotation information.

Fig. 8. The extracted information in OEM format.
Similarly, the web data about trading volume of the light sweet crude oil futures can also be extracted according to the above extraction process. However, this process is only to extract information on the web and it cannot integrate the information from different sources. In order to fuse these data from the web and construct a web warehouse, the mediation services and mapping service can be used. Fig. 9 shows an overall EFML process of web warehouse construction.

In this example, we have two groups of web pages. In the extraction process, the information about quotation of crude oil futures in the first group is extracted by the specific wrapper service or extractor program. In the second group, information about trading volumes can also be extracted in the similar way. In the fusion and mapping processes, possible data conflicts between different web pages of the same group can be solved by m mediator and mapping services. Similarly, the fusion between the groups can also be performed by M mediator and mapping services and thus an integrated schema can be obtained by fusing two sub-schemas. In the integration process, the mediators mainly use the equivalence correspondence to

---

Fig. 9. The web warehouse construction example with the EFML process model.

Fig. 10. The dimension model for the crude oil future quotation-trading example.
integrate different information. For example, the $M$ mediator uses, among others, the equivalence correspondence between time of sub-schema 1 and time of sub-schema 2 to perform the information fusion. It contains the relation quotation-trading with attributes time, last, high, low, changes and volume. Actually, the integrated schema creation in this example is based on a dimension model, which is shown in Fig. 10. Usually, a dimension model is a logical design technique that seeks to make data available to end users in an intuitive framework to facilitate querying. Finally, the corresponding web warehouse relation quotation-trading also contains the attributes time, last, high, low, changes and volume.

4. Conclusions

In this paper, a new web information fusion tool – web warehouse, which is suitable for web mining and knowledge discovery is proposed. In the proposed web warehouse, a layered web warehousing architecture for decision support is introduced. In terms of the four-layer web warehouse architecture, an EFML process model for web warehouse construction is then proposed. In the web warehouse process model, a series of web services, including wrapper services, mediation service, ontology service and mapping service are used. Particularly, two kinds of mediators are introduced to fuse the heterogeneous web information. Finally, an illustrative example is presented to interpret the web warehouse construction process. The experiment implementation process implies that such a web warehouse can not only increase the efficiency of web mining and knowledge discovery on the web, but also provide an effective web information fusion platform.

Acknowledgements

The authors would like to thank the Editor-in-Chief, three guest editors and four anonymous referees for their valuable comments and suggestions. Their comments helped to improve the quality of the paper immensely. The work described in this paper was partially supported by the National Natural Science Foundation of China (NSFC No. 70601029), the Chinese Academy of Sciences (CAS No. 3547600), the Key Research Institute of Humanities and Social Sciences in Hubei Province-Research Center of Modern Information Management, and the Strategic Research Grant of the City University of Hong Kong (SRG No. 7001806).

References


