A Framework for Developing the Web-based Data Integration Tool for Web-Oriented Data Warehousing

PATRAVADEE VONGSUMEDH
School of Science and Technology
Bangkok University
Rama IV road, Klong-Toey, BKK, 10110, THAILAND
patravadee.v@bu.ac.th

ANON SUKSTRIENWONG
School of Science and Technology
Bangkok University
Rama IV road, Klong-Toey, BKK, 10110, THAILAND
anon.su@bu.ac.th

Abstract— **Since the data relate to the Data Warehouse (DW) would be extracted from the heterogeneous data sources, the tool used for integrating the disparate data should operate carefully. Moreover, regardless of the operating location and the limitation of technical requirement on the client-side application, the certain tool should enable the DW administrators to perform the data integration process easily and flexibly. Therefore, the purpose of this research is to develop the prototype of "the Web-based Data Integration Tool for Data Warehousing." To achieve this purpose, the Java Servlet and the Hypertext Markup Language (HTML) are selected to construct the prototype. After system construction and testing, the semi-structured interviews are set up at the four supportive organizations. The results show that the prototype effectively and efficiently provides the data integration supports to the DW administrators. Moreover, the concept of the "Thin Client" is achieved.**

Keywords—Data Warehousing, Data Integration Tool, Web-based Data Integration, Thin Client

I. INTRODUCTION

In the era of the business competition, the business situation is changing rapidly. It is characterized by an increasingly dynamic business environment, high demand for current, accurate, and integrated information used for supporting the business activities. It is well known that the high business management skill is very important for managing the rapid change of business situation. The more we understand the business, the more complex business situation we can achieve.

Nowadays, the computing technology and the business management skill; however, are converged to come up with the more efficient, effective, and successful business operation. One of the very popular computing technologies that provide the capability of responding the rapid change of the business situation is the "Data Warehouse Technology". As a technology represented on the basis of an integrated data source, the data warehouse must be built carefully. The data integration process, that influences not only the availability, but also the quality of data in the warehouse, must be performed properly by support of the "Data Integration Tool." This tool facilitates the data warehouse administrator in performing the data integration to serve across the enterprise. If the Data Warehouse is not represented on the basis of an integrated data source, they can be ineffective and can be disastrous. Therefore, it is the challenge of the data integration process and tools to reduce the disparate data and create an integrated data that would be used for supporting the business information demands.

Since the World Wide Web is converged to all types of concept and tool to come up with the higher system performance, availability, and flexibility. Therefore, this research focuses on performing the data integration through the web by using the thin client. Working with the combined concepts of thin client and web leads to the development of the *"Web-based Data Integration Tool."* By the support of this tool, the data warehousing can be performed easily and flexibly through the web by using the thin client.

In fact, the purpose of this research is not to replace the existing client-server products or tools provided by the current vendors (e.g., ETI –Extract, IBM: InfoSphere DataStage, Informatica: Cloud Data Integration, Oracle: Data Integrator) with the web-based tool. But, it intends to propose another possible, flexible, and cheap alternative for integrating the heterogeneous data through the web.

II. THE STATEMENTS OF THE PROBLEM

Though the existing predominant data integration tools provided by the current vendors can gracefully support the data integration process, the given tools do not completely provide the flexibility and availability of performing the data integration process. These tools are still working based on the "full client" concept. Therefore, the data integration process can be performed by only some client machines which meet the technical requirement (e.g., disk space, memory, operating system) of the tools. Moreover, some of the current predominant data integration tools are binding with some types of DBMS, such as MySQL, DB2, Oracle, Sybase. This characteristic may turn to be the stumbling block of the further redesign or adjusting of the IS infrastructure, such as shifting from two-tiered client/server to three-tiered client/server. From this perspective, the three major disadvantages of the current data integration tools are:

- The inflexibility and unavailability of performing the data integration caused by the full-client requirement of the tool.
- The data integration process and the Data Warehouse manipulation are limited by the location, in which the client-side application is installed.

- The difficulty of the further system redesign caused by the DBMS binding of the tool.

III. The Features of the Predominant Tools

The data warehouse is the repository of integrated data extracted from distributed, autonomous, and heterogeneous sources. The data is extracted from many sources, integrated with data from other sources, and loaded into the data warehouse. The users focus on different subject areas then access and analyze the data through the warehouse.

The top players and products in data integration area (e.g., ETI Extract, IBM InfoSphere DataStage, and Oracle Data Integrator) facilitate and pay more attention to the "continuous updating" of the enterprise data warehouse. However, these dominant data integration tools are usually relied on the platforms. That is, these tools automate the integration process by using a workstation client (with Graphical User Interface) to control the data integration process. Some of the predominant data integration tools are described as follows.

1) WHIPS (WareHousing Information Project at Standford):

The WHIPS prototype focuses on the data integration component. It identifies the data changes at the heterogeneous sources, transforms them, summarizes them, and integrates them into the warehouse. Each source involved in the warehousing system will be manages by the internal module, called "monitor". The monitor is responsible for detecting modification on the source data and notify the integration of them. The "wrapper", one of the important component, translates the single source queries to the queries in the native language of its source.

From the stated features, the warehousing system is allowed to deal with many DBMSs. The modifications in and accessing to each DBMS can be managed dynamically and automatically. The wrapper thus shields all other modules in the WHIPS system from the particulars of the warehouse, and allows any database to be used as the warehouse [1].

2) ETI Extract:

EIT Extract, a "client/server product", targets data conversion/transformation and bridging in heterogeneous environments. EIT Extract provides developers with a GUI to generate extraction programs. The data warehouse administrators are allowed to move data through simple point-and-click interaction. The data can be imported from many DMBSs (e.g., DB2, Oracle, Sybase) and exchanged between incompatible systems. The monitoring function of ETI Extract is script-driven, which enables the data warehouse administrators to schedule and monitor data feeds through the scripts [2].

3) IBM InfoSphere DataStage:

IBM InfoSphere DataStage, another GUI client/server ETL product, enables the data warehouse administrator to develop the parallel jobs for connecting, extracting, transforming, and writing data to a target database/data warehouse or file. Since the high quality data lead to the accurate information and help decision makers to make good decision on their businesses, so the data cleansing process could be performed by the support of another add-on module called InfoSphere QualityStage [3].

IV. The Proposed Solution

To implement the web-based data integration tool that completely support the concept of the thin client and enables the Data Warehouse administrator to easily perform the data integration process across the web, the "Java Servlet Technology" (or servlet) is selected to implement the given tool. The servlet is the technology that completely fulfills the concept of thin client. To work with the servlet, client interacts with the HTML forms and sends the data integration requests to the server through that forms. After receiving the requests, the servlets operate the request solely within the domain of the server. Then, the results are sent back and browsed on the client side.

By support of the JDBC API and the defined set of generic SQL types that represent the most commonly used SQL types, servlet is able to send SQL statements to the appropriate databases. Therefore, the developer will be able to use these JDBC types to reference the generic SQL types, without concerning about the exact SQL type name used by the target databases.

V. The Research Methodology

The research methodology is divided into 4 phases as follow:

- ***Phase 1: Study of the data integration process, techniques, and tools.***

By analyzing and trying out the predominant existing data integration tools [1,2,3,4,5] and reviewing the literatures relate to the Data Warehousing Technology [6,7,8,9], the characteristics of the tools used for supporting the data integration process are identified. Moreover, the workflows or the stages of data integration process are revealed [10,11]. Many researches show that "the Web Technology" enables all stages of the Data Warehousing process to be performed easily, efficiently and flexibly thru the web. [9,12]

- ***Phase 2: Analyzing and design the system prototype.***

The data integration process in the context of banking system and insurance systems are analyzed and designed in order to identify the necessary system components and modules. The target user of the proposed prototype is "the DW administrator" of the supportive organizations. After identifying the system's target user, the in-depth interviews are set up at the supportive sites. The interview result revealed the characteristics of the integrated data and, again, the workflow of the data integration process. As shown in Fig.1, the workflow within the framework of the *"Web-based Data Integration Tool"* is divided into five steps, which are:

Step 1: The client identifies the request and then sends the concerned DB information (e.g., DB Location, DBMS type, DB Name, SQL statement) to the server running servlets.

Step 2: The servlet running on the server will parse the SQL statement to point out the involved source and the target DB that the integration process (or SQL statement) has to deal with.

Step 3: By the support of the JDBC API, The server establishes the database connection with the data source and the target database based on the information received from Step 1. If the connection failed, the error message would be sent back and browsed on the client. Otherwise, the request would be responded.

Step 4: The servlets, which control the integration process, send the parsed SQL statement to be executed properly on the related databases.

Step 5: After executing SQL statement, the results of the integration process are sent back and browsed on the client in the form of HTML content.

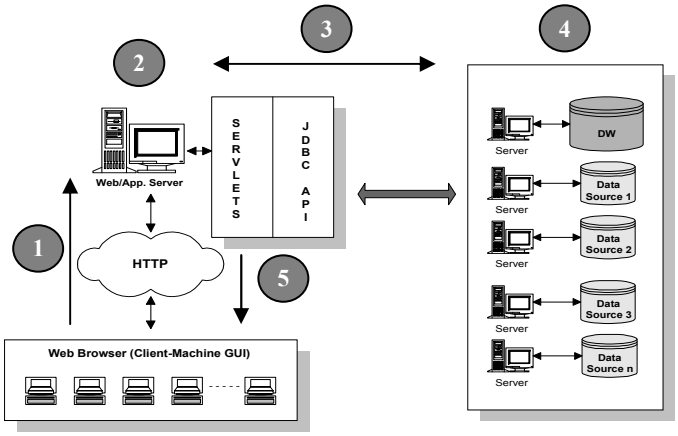

Fig.1 The Framework of the Web-based Data Integration Tool

To fulfill the data integration process, the proposed prototype provides three main modules. These modules are:

❑ *The Flat-File Integration*

This module is responsible for integrating the flat-file with the target database or data warehouse. Working with this module, the personal data of the executives, the data from external sources, and the data archived from the legacy system can be integrated with the data warehouse easily. For the contents of the flat-file, each row of the flat-file represents data record to be integrated with the data warehouse or target database, while substring placed in between by the delimiter represents data field. For example, the content of the flat-file can be

2330233473|'Teera Soontorntai'|'101 Rama IV Road, BKK'|'M'|28500

, where

2330233473 represents Account Number,
'**Teera Soontorntai**' represents Account Name,
'**101 Rama IV Road, BKK**' represents Customer Address,
'**M**' represents Customer Gender, and
28500 represents Account Balance.

The working steps of the flat-file integration module are:

1. Identify the target database and flat-file, which contains the archived data, to be integrated.
2. Establish the database connection based on the provide information.
3. Open the identified flat-file.
4. Read contents of the flat-file each row a time.
5. Fetch the data field from the data row read by step 4.
6. Create SQL statement (i.e., "INSERT INTO Table-Name VALUES(*fetched data field*);").
7. Execute the SQL statement created by step 6.
8. Repeat step 4 until there is no more data, end of file.
9. Close the flat-file and the database connection.

❑ The DB Integration

The second module is the database integration module which is responsible for integrating the legacy data stored in the operational database with the target database (or data warehouse). This can be done by extracting the legacy data directly from the operational DBMS, and, then, transforming and loading the data into the target database. The working steps of the DB integration module are:

1. Identify source database and target database (or DW) to be connected.
2. Identify SQL script file to be used.
3. Check types of DBMS (e.g., Oracle, Informix, Sybase, DB2, MySQL, etc.) to be connected with and, then, establish the database connections.
4. Open the identified SQL script file.
5. Read the script file's contents and, then, fetch each SQL statement a time.
6. Parse the SQL statement in order to properly point out the database (data source or target database) that the sub-SQL statement would be sent to execute.
7. Execute the SQL statement on the relevant data sources and target database.
8. Repeat step 5 until there is no more SQL statement.
9. Close the SQL script file and the database connection.

The example of data integration supported by this module is shown as follows (Fig.2 and Fig.3).

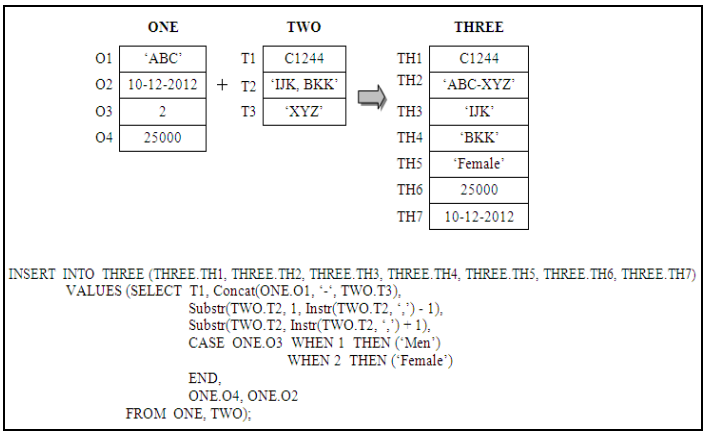

Fig.2 The Example of Data Integration

For performing the data integration as shown in Fig.2, the data integration rule, shown in Fig.3, must be pursued.

Data Integration Rules:

THREE.TH1 = TWO.T1
THREE.TH2 = ONE.O1 + TWO.T3
THREE.TH3 = TWO.T2 (Strings before comma (,))
THREE.TH4 = TWO.T2 (Strings after comma (,))
THREE.TH5 = IF ONE.O3 = 1 THEN THREE.TH5 = 'Male'
ELSE THREE.TH5 = 'Female';
THREE.TH6 = ONE.O4
THREE.TH7 = ONE.O2

Fig.3 The Example of Data Integration Rule

❑ The DW Manipulation

The last important module is the DW manipulation module which provides the capability of manipulating data stored in the warehouse. This capability facilitates the DW administrator in restructuring or modifying the data warehouse structure (or the schemas in the data warehouse). The data manipulation can be creating schema, dropping schema, modifying some data fields, and so forth.

Since the SQL scripts working with this module consist of SQL statements that would be executed on single data source, the SQL parser is not required. Moreover, regardless of SQL syntax, the JDBC API provides the capability of sending the SQL statement to be executed on the related DBMS properly. The various syntax of SQL statement, hence, can be used with this module. However, the SQL syntax must be interpreted and accepted by the affected DBMS. The working steps of the DW Manipulation module are:

1. Identify the target database (i.e., the data warehouse) to be manipulated.
2. Identify the SQL script file to be used.
3. Establish the database connection.
4. Open the SQL script file.
5. Read the file contents and, then, fetch each SQL statement a time.
6. Execute the fetched SQL statement.
7. Repeat step 5 until there is no more SQL statement.
8. Close the SQL script file and the database connection.

❑ ***Phase 3: Constructing and testing the prototype.***

After analyzing the work flow of data integration process and identifying the components and modules necessary for performing the data integration, the prototype is constructed. The system's modules are implemented in Java Servlets. While, the E-forms and the results represented on the client side are implemented in HTML and Java Script (as shown in Fig.4 to Fig.7). The prototype is, then, tested. Three DBMSs, which are "Oracle", "DB2", and "Informix", are selected for testing in order to evaluate the correctness of the data integration workflow, efficiency, and effectiveness of the prototype.

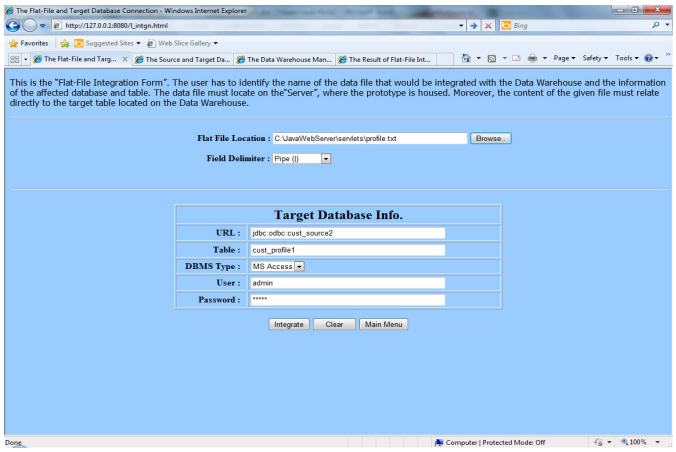

Fig.4 The Flat-File Integration Module

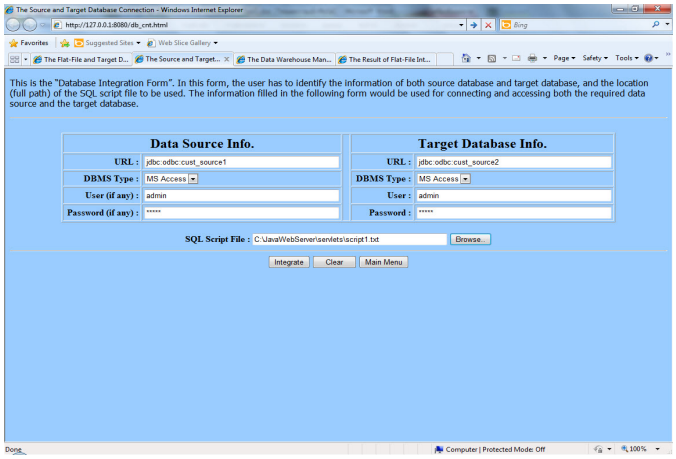

Fig.5 The DB Integration Module

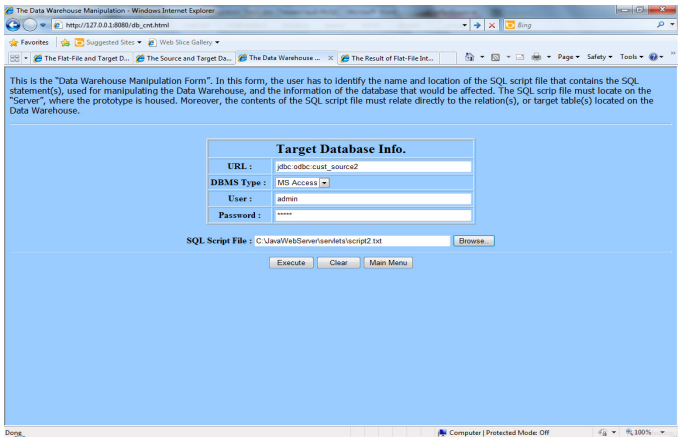

Fig.6 The DW Manipulation Module

Fig.7 The Results Represent on the Client-Side

❑ ***Phase 4: Evaluating the prototype.***

After system testing, the prototype is installed on the server of the supportive organizations (two banks and two insurance companies). The data warehouse administrators of these organizations are requested to try working with the prototype. Then, the semi-structured interviews are set up at the supportive organizations in order to evaluate the prototype. The evaluation results are:

1. The prototype facilitates the DW administrator to perform the integration process easily and flexibly (no location restriction).

2. The concept of "Thin Client" is achieved. That is, no special technical requirement is required on the client machine.

3. The simple GUI browsed on the web browser enables the user (or the data warehouse administrator) to work with the prototype intuitively.

4. By working with the prototype, however, the DW administrator has to deal with the burden of writing raw SQL.

VI. CONCLUSION

As a result of study, the data warehouse administrator can perform the more flexible data integration with the web-based data integration tool. To perform the data integration through the web, E-forms are browsed by the web browser on the client machine, and, then, all requests are sent by the client machine to the server, where the prototype is housed. After receiving the requests, the server processes those requests and then responds back to the client who sent the requests. These responses are in the form of HTML contents which can be easily browsed on the client machine. The SQL statements, stored in the SQL script file, play an important role in extracting, transforming, and loading data from the related data sources. Working with the script driven tool, the data warehouse administrator can modify the existing SQL script and reuse that script to conduct data integration in the future. Moreover, to make it simple, the prototype allows the data warehouse administrator to conduct the data integration by using the standard SQL syntax which can be interpreted by the relevant DBMSs. The user is not requested to learn any new programming language or tool, except the related SQL statement.

VII. THE FURTHER STUDY

The further studies of the research can be done in many areas as follows:

1. To create "the Complex SQL Parser" for supporting the complex/nested SQL statements.

2. To create "the Web-based Metadata" for facilitating the DW administrator in navigating the data in the DW easily.

3. To adjust the prototype in order to provide more support on the DBMS types.

4. To provide the capability of continuous updating for the web-oriented data warehouse.

REFERENCES

[1] J. Wiener, H.Gupta, W. Labio, Y. Zhuge, H. Garcia-Molina, and J. widom, "The WHIPS Prototype for Data Warehouse Creation and Manitenance," The 13th Int. Conf. on Data Engineering, U.K., 1997, pp. 589 – 590.

[2] ETI (2012). (2012). *ETI Product Details.* [Online]. Available: http://www.eti.com [Cited 1 February 2012].

[3] IBM Corporation. (2012). *IBM InfoSphere DataStage.* [Online]. Available: http:// www-01.ibm.com/software/data/infosphere/datastage/ [Cited 1 February 2012].

[4] Informatica Corporation. (2012). *Cloud Data Integration: Access and Integrate Cloud-based Data with On-promise Sources*. [Online]. Available: http://www.informatica.com/us/products/cloud-data-integration/ [Cited 1 February 2012].

[5] Oracle Corporation. (2012). *Oracle Data Integrator*. [Online]. Available: http://www.oracle.com/technetwork/middleware/data-integrator/overview/index-088329.html [Cited 1 September 2012].

[6] C. Imhoff, N. Galemmo, and J. G. Geiger, *Mastering Data Warehouse Design: Relational and Dimensional Techniques*, Indianapolis, IN: Wiley Publishing, Inc., 2003.

[7] W. H. Inmon, *Building the Data Warehouse*, 4th ed., Indianapolis, IN: Wiley Publishing, Inc., 2005.

[8] R. Kimball et al., *The Data Warehouse Lifecycle Toolkit: Practical Techniques for Building Data Warehouse and Business Intelligence Systems*, 2nd ed., Indianapolis, IN: Wiley Publishing, Inc., 2008.

[9] R. Kimball, R. Merz, *The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse*, New York, NY: John Wiley & sons Inc., 2000.

[10] T. A. Bennett, and C. Bayrak, Bridging the Data Integration Gap: From Theory to Implementation, *ACM SIGSOFT Software Engineering Notes,*Vol.36, No.4, 2011, pp. 1-5.

[11] S. H. A. El-sappagh, and A. M. A. Hendawi, A Proposed Model for Daa Warehouse ETL Processes, *Journal of King Saud University – Computer and Information Sciences,* Vol.23, 2011, pp. 91-104.

[12] X. Tan, D. C. Yen, and X. Fang, Web Warehousing: Web Technology Meets data Warehousing, *Technology in Society,* Vol.25, 2003, pp. 131-148.