ICEB 2010 Proceedings

International Conference on Electronic Business (ICEB)

Winter 12-1-2010

The Research on ETL of Web Data Source from Postal Enterprise IS

Huaichang Hou

Jihui Shi

Lanjuan Liu

The Research on ETL of Web Data Source from Postal Enterprise IS

Huaichang Hou, ShanXi Institute of Economic and Business, Shanxi, China.
Jihui Shi, School of Information Management and Engineering, Shanghai University of Finance and Economic, Shanghai, China.
Lanjuan Liu, Shanghai University of Finance and Economic, Shanghai, China.
E-mail: 18935131368@189.cn, keaide1@139.com, lljuan@mail.shufe.edu.cn

Abstract

In this paper, based on an actual problem in a process of data warehouse project which an enterprise implemented, we make a study on the ETL framework of Web data source from Enterprise IS which is B/S structural, and hence presented a new extraction and loading method for the situation without directly database connection. Furthermore, we developed a new ETL framework with the ability of processing Web page as data source in B/S structural enterprise IS. With the extension of XML configuration file, the framework can be adjusted freely along with business change.

I Introduction

Data is the core of data warehouse. To ensure the data can be loaded integrally, accurately into data warehouse, the completely ETL process is the key of data warehouse project being successfully implemented. [1]

The currently research of ETL mainly focus on the fusing of pattern projection, data cleansing and data transform along the data transformation process, and how to trace the source of the data in warehouse by using the description of ETL process. The related study mainly include: Clio System developed by Toronto University [2], focused on project among each kind of patterns in data transformation; the INRIA institute of French achieved the logical description of data cleansing corresponded data transformation program in the data cleansing system, and implemented the physical execution with different approach [3]; in the Potter's Wheel system , a data cleansing system developed by Berkley university, a completely algebra description of data cleansing was brought about [4]; etc.

On the stream of the research of framework, Y. L. You and X. Zhang [5] provided a ETL framework can be easily extended. This framework uses a file as the mediate between data warehouse and its data source. Then the couplings among functions were reduced. In the paper [6], H. G. Chou and J. C. Chou designed a wholly ETL framework from the point of practical view. In the framework, a common access interface was created to shield the discrepancy among all kind of data source, and offered a common and effective scheme to eliminate mode and data collision of multi-data source. The paper [7] designed a data increasable ETL framework which integrated multiple capture functions to eliminate the discrepancy of capturing data from data source with different structural. Iqbal T and Daudpota N. [8] designed the whole ETL process by using XML, in order to make the ETL process more extensive and can handle data with different structural more effectively.

The above research partly settled the data extraction and transformation problem from different data source, and improved the performance of ETL tools. But there are still some situations not be concerned, such as when we regard the Web page generated by B/S structured enterprise IS as ETL data source. In this paper, we will focus on this. From a practical case of an data warehouse program of a certain postal company, a department controlled data warehouse cannot directly connect database of B/S structural IS belong to some other department, by study the process of extraction, transformation and load of its Web page data, we find a way and designed a framework, and at last we achieved expected consequent with the success of the implementation of the program. At the same time, we further completed the framework and potentiate it more extensive characteristics. That means it can works smoothly with the change of business. Even it can be used in other areas after proper modification, such as the ETL problem of Web community data warehouse.

II Background of the Research

In an enterprise, a department or its sub-company using a B/S structural business system and its user query and use data by Internet browsers. With proper authorities, the database provides correspond data to users. Another department or its sub-company plans to build data warehouse and need the data support from the department or sub-company previous. But, for the reasons of the whole enterprise system and evolution history, their IS are independent, and can not access freely to each other by using bottom connection of database. So, the data warehouse project countered complicated problems. Because the ETL tools the data warehouse product integrated has no the capability which can settle it effectively. For this awkward situation, someone think that Web Spider can capture the data from browser without bottom connections from that department's business IS database. But generally, Web Spiders designed mainly for usual Internet websites and capture ordinary Web page information, and the Web pages B/S structural enterprise IS generated are for business, and very different from usual pages. So, general Web Spider cannot capture information from the systems. Such as, to get any information must be authorized first, the data only can be read through some manual operations, and the most important, the structure of the reports displayed on the browser are varied, complicated, operation related and numerous business information.

So, in order to deal with the problem, we find a way and designed a ETL framework in this paper. In section 3, we will present the framework and the method. In section 4, we introduced a case to show the framework and the method has already been successfully used in practical.

III The ETL Framework of Web Data Source from Enterprise IS

The whole ETL framework composed with the followed several parts (Figure 1):

1. the Web data show layer
2. the template for data extraction (some configuration files written by XML)
3. the Web data capture module
4. the data load module
5. the data import template

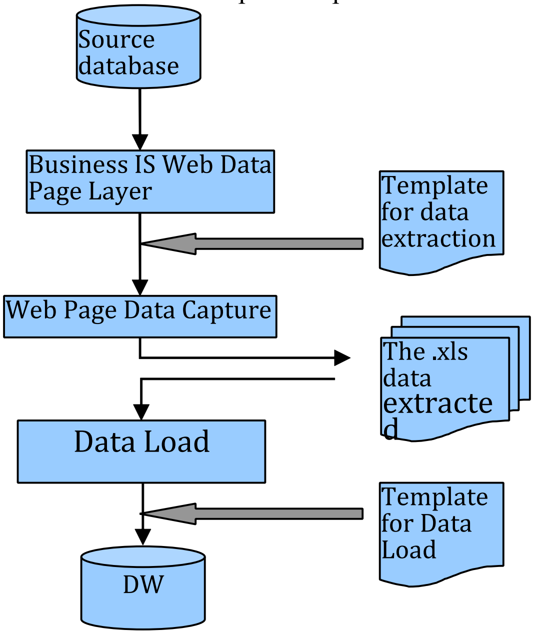


Figure 1 The ETL framework for Web data source

Unlike the usual Web Spider, our framework has to additional parts, the data extraction template and data import template. The both are written by XML and make the framework can adapt to more business information. By using the XML configuration files, the level of couplings among each module of the framework are lowered. Each module can works more independently under the coordinate of the configuration files for the whole ETL process.

1. The Web data show layer
   The layer means each operational system which provides data for the data warehouse or the Web data source. Include all Web pages embodied report data. Usually, the source will contain tens, even about one hundred, reports distribute in every functional modules, and the browser access data with URL and parameters.
2. Data extraction template
   These templates are to assist the capture module to extracts data from the Web data show layer. Though report pages in Web data show layer have very different structures, in different position, in fact its' only URLs with parameters for browser. Browser submits the URL to server and the server response the request and returns HTML page embodied data we required. In order to make the capture module ignores URLs of each page, structures or other information of reports, we use XML files to configure these information. The following shows the structure of a typical report page template.

```
<!--<Report index="1">
        <ReportClass desc="the class of
report">card business infor</ReportClass>
        <ReportName desc="report
name">XX business data
collect-publication-city
degree</ReportName>
        <ConfigFileName> XX business data
collect-publication-city
degree.xml</ConfigFileName>
        <DataType desc="date
type:1monthly.2.daily">1</DataType>
        <ReportType desc="report type.
1,standard; 2,">1</ReportType>
        <DptName desc="the name
region"></DptName>
        <Url desc="Report
Url"><![CDATA[http://plserver/reportsearch/
repor?....]]></Url>
</Report>
```

<Report> defines the mark of the page to be extracted; <Class> defines the classification of pages to classify and manage large mount of pages to be extracted; <ConfigFileName> defines the related import template when the page be imported into data warehouse; <DataType> defines the date mode of the extracted page, such as daily report, monthly or seasonal etc. <Url> defines the URL of the

page to be extracted, and this is the most important point. The extraction module submits this mark to IS server and receive its response.

3. The Web page extraction module

This module is the main part of the whole framework. By simulating users logon the system and the interaction mechanism, the data of the IS database can be read like a user is accessing and interacting with the system through browsers. Then with the Web page extraction template, the page was extracted and saved as Excel files. The program pseudo-code is as follows:

```
  WebSpider ( )
        {    Logon.set(URLlog, UserID,
Password, Period)
           Readpage.statues(Browser, Scripts,
        Contrles)
         Logon(connect)

        TempleteDoc.load("LsTemplete.xml")
        Windowmanager.ActiveBrowser.Navi
gate(URLmanage)
        PageFrameset()
        WindowManager.
ActiveBrowser.Document.InvokeScript("login
Manage")

        Windowmanager.ActiveBrowser.Navi
        gate("URLsearch")
        searchAreaElement.setLinks()
        WindowManager.ActiveBrowser.Navig
ate("URLsearchresult");

        Readpage.statues(Browser, Scripts,
Contrles)
         SetContrlerules()
         ReportStructureet()
         ReportDownload()
         setDate()

        DataSave("dir\filename.xls")    }
```

4. The data load module

This part runs follow the extraction module. Checking all page Excel files extracted from the IS database, capture and transform the data to the form we wanted and load it to data warehouse. Because the Excel files which produced by the extraction module have varied topics, different structures, ETL tool cannot process them directly. So, slef-developed program and templates (introduced in the next part) for each kind of files are needed. The templates are used to identify data cells in page Excel file. Below are two c# functions (the outline) of the module to read data import template (the LsTemplate.xml template) and capture the data from associated page Excel files.

```
  private static string GetItemID(int Row, int
Col)        //reading template and return data
from non-data area
  {    XmlDocument doc = new
XmlDocument();

doc.Load("LsTemplete.xml");
          XmlNode node =
doc.SelectSingleNode("/LSTemplate/GetItem
ID/Node[Row=" + Row.ToString() + "][Col="
+ Col.ToString() + "]");
           if (node != null &&
node.ChildNodes[2].InnerText.Length > 0)
          { return
node.ChildNodes[2].InnerText.ToString(); }
                         return ""; }

   private static System.Data.DataTable
GetItemID()     //return data from data area
   {    XmlDocument doc = new
XmlDocument();

doc.Load("LsTemplete.xml");
          ...          //define the rows
           XmlNodeList nodes =
doc.SelectNodes("/LSTemplate/GetItemID/N
ode");
           foreach (XmlNode node in nodes)
           {    XmlNode row =
node.SelectSingleNode("Row");
           XmlNode col =
node.SelectSingleNode("Col");
           XmlNode itemid =
node.SelectSingleNode("ItemID");
           DataRow dr = dt.NewRow();
           dr["Row"] = row.InnerText;
           dr["Col"] = col.InnerText;
           dr["itemid"] = itemid.InnerText;
           dt.Rows.Add(dr); }
           return dt; }
```

5. The data import template

In order to deal with the situation of varied topics and structures of page Excel files for the data load module, we setup some configuration files for different file types by XML. In the configuration file, the content of the page Excel file was described to a 2-D structure, so that the data loading module can distinguish the cells. In this way, whatever the business system, norms or report structure changes, we only need to rectify the parameters of configuration files, that means the program is totally divided with data. A typical example is as follows.

```
  <ImportTemplate>   <!—define data range.
Defining non-data area such as headers or
footers of Excel table.-->
     <Region>
     <StartRow></StartRow>
<StartCol></StartCol>
     <EndRow></EndRow>
     <EndCol></EndCol>
      </Region>
     <Rules>   <!—defining operation rules.
Such as the rule 1 used to transform
```

```
proporation, the rule 2 used to transform unit.
Can define many rules following business. -->
            <Rule Name="rule 1">
            <OperFlag>+</OperFlag>
<OperValue>1</OperValue>
            </Rule>
            <Rule Name="rule 2">
            <OperFlag>*</OperFlag>
<OperValue>10000</OperValue>
            </Rule>
           </Rules>
           <Data>
            <Node>    <!—a node means a cell
in Excel table -->
            <!—this node represents the
attributes of cell C1. including the unit, index,
data type, etc. such as some post office, newly
added savings, real data value, etc. -->
            <Row>1</Row>       <Col>3</Col>
            <ItemID>Norm-ID</ItemID>
            <DptID>Unit-ID</DptID>
            <Type>Data type. Such as real value
or budget value, or other attributes. </Type>
            <Rule>rule 2</Rule>
            </Node>
                ….
            <!—continue the definition. Can
define the whole row or column or data block.
>
   </Data>
</ImportTemplate>
```

The template above describes the definition of non-data area and its' rules in a configuration file. Then take a C1 cell as example to express the meaning of data in this cell, such as which agency it belongs to, which norm it adapts, or if it needs transformation. In this way, we even can define where the data in this cell was saved in database table, or define the whole column and row or data block.

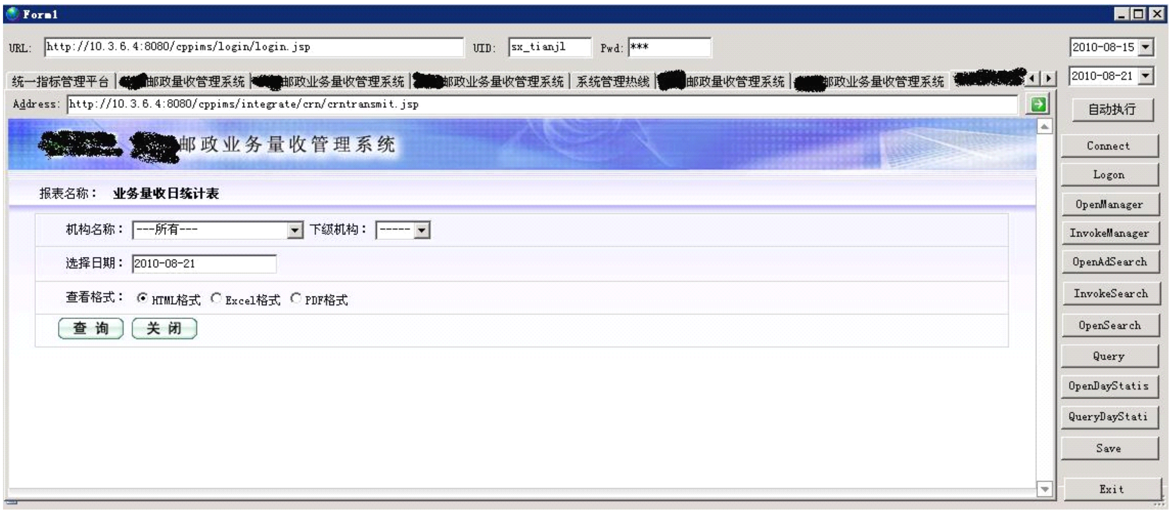
IV Application Case

A state run province degree postal company in China would built a data warehouse in its general headquarter to integrate all resources in the province to provide DSS support for each management level. But, for the reasons of the whole enterprise operation system and its special evolution history, sub-companies of every city in the province all had build their own B/S structural business systems, and these systems cannot be accessed at the database level to general headquarter. In this case, the data warehouse product which they purchased cannot deal with this situation to extract data from those databases by using the integrated ETL tool. In order not to affect the implementation of the whole project, they had to extraction data manually in the early stage to support the development process, and only can cover little part of items because of the low efficiency of handwork. So, in the early stage, the powerful ability of the whole system was greatly limited, and the error rate was very high because of the manual data import work. After developed a new ETL tool with the framework and method this paper depicted, the data extraction can worked smoothly and automatically, the efficiency was greatly improved and covered all business data item in the province, and except some temporary operation data norms (For the reason of some business operation, there would be some norms temporally added in reports.) the fixed data items have not meet any fault compared the early stage.

Table 1 Manual and ETL

Index Mode	Error Rate (/100)	Time cost	Norms Covered
Manual	3—8	Hours	20%
ETL	0.1—0.3	Several minutes	95%

The ETL tool mainly include the following functions:


Form1
统一指标管理平台
[redacted]邮政量收管理系统
[redacted]邮政业务量收管理系统
[redacted]邮政业务量收管理系统
系统管理热线
[redacted]邮政量收管理系统
[redacted]邮政业务量收管理系统
[redacted]
[redacted]邮政业务量收管理系统
查 询
关 闭
自动执行
Connect
Logon
OpenManager
InvokeManager
OpenAdSearch
InvokeSearch
OpenSearch
Query
OpenDayStatis
QueryDayStati
Save
Exit

URL: http://10.3.6.4:8080/cppims/login/login.jsp UID: sx_tianjl Pwd: ***

2010-08-15

2010-08-21

Address: http://10.3.6.4:8080/cppims/integrate/crn/crntransmit.jsp

报表名称： 业务量收日统计表

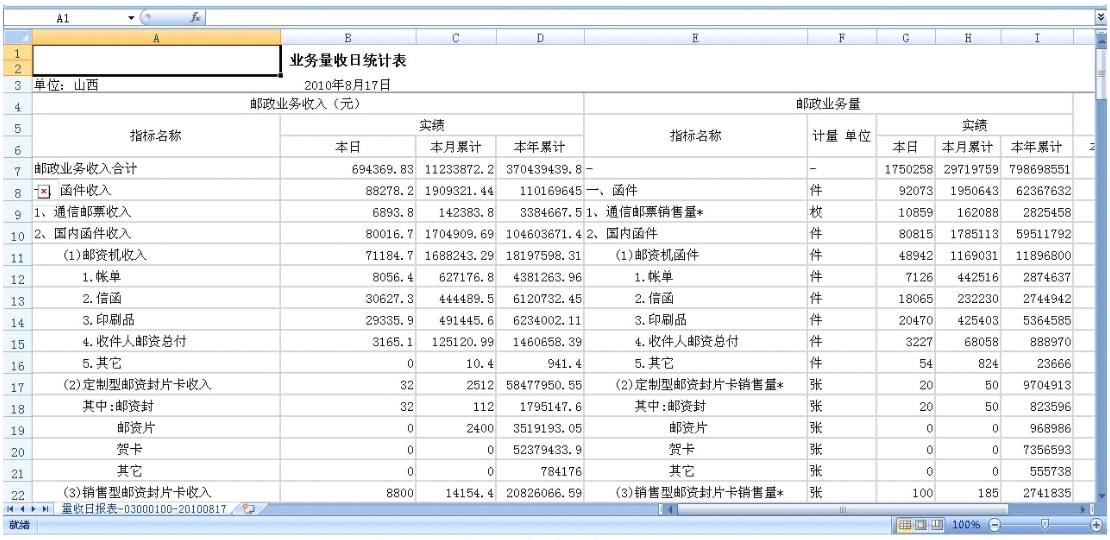
机构名称： ---所有--- 下级机构： -----

选择日期： 2010-08-21

查看格式： ● HTML格式 ○ Excel格式 ○ PDF格式

Figure 2 the query interface of business reports

simulating the action of logon by user and navigating to the report page which displays the daily business data, then the Web page data extraction module extracts all cities' business data of the whole province using data extraction template and save the page file as Excel files to local, then the data import module load the data in these Excel files into data warehouse under the support of data import templates.



业务量收日统计表

单位：山西 2010年8月17日

	邮政业务收入（元）				邮政业务量				
	指标名称	实绩			指标名称	计量 单位	实绩		
		本日	本月累计	本年累计			本日	本月累计	本年累计
7	邮政业务收入合计	694369.83	11233872.2	370439439.8	-	-	1750258	29719759	798698551
8	函件收入	88278.2	1909321.44	110169645	一、函件	件	92073	1950643	62367632
9	1、通信邮票收入	6893.8	142383.8	3384667.5	1、通信邮票销售量*	枚	10859	162088	2825458
10	2、国内函件收入	80016.7	1704909.69	104603671.4	2、国内函件	件	80815	1785113	59511792
11	(1)邮资机收入	71184.7	1688243.29	18197598.31	(1)邮资机函件	件	48942	1169031	11896800
12	1.帐单	8056.4	627176.8	4381263.96	1.帐单	件	7126	442516	2874637
13	2.信函	30627.3	444489.5	6120732.45	2.信函	件	18065	232230	2744942
14	3.印刷品	29335.9	491445.6	6234002.11	3.印刷品	件	20470	425403	5364585
15	4.收件人邮资总付	3165.1	125120.99	1460658.39	4.收件人邮资总付	件	3227	68058	888970
16	5.其它	0	10.4	941.4	5.其它	件	54	824	23666
17	(2)定制型邮资封片卡收入	32	2512	58477950.55	(2)定制型邮资封片卡销售量*	张	20	50	9704913
18	其中:邮资封	32	112	1795147.6	其中:邮资封	张	20	50	823596
19	邮资片	0	2400	3519193.05	邮资片	张	0	0	968986
20	贺卡	0	0	52379433.9	贺卡	张	0	0	7356593
21	其它	0	0	784176	其它	张	0	0	555738
22	(3)销售型邮资封片卡收入	8800	14154.4	20826066.59	(3)销售型邮资封片卡销售量*	张	100	185	2741835

Figure 3 the result after the data extraction

Because we used the XML configuration file to reduce the coupling between each module, we don't need to rectify the program when business changed. What we need to do is just to modify the correspond variables. The below are some pictures captured from the ETL system.

V Summary

For many types of reasons, many organizations will counter many complicated problems of accessing and extracting data from old IS systems when implementing data warehouse. These specific problems usually would not be predicted and included by general ETL tools, or become the mainstream of research to be totally solved. Such as the situation we depicted it in this paper, the general ETL product doesn't provide complete solutions for this kind of unusual situation. Through the research in the paper, we offered a complete solution for this kind of problems. And the flexibility of the solution makes it even can be used in other domain, such as the out sourced data extraction of Web community research.

References

[1] Willian H.Inmon. Building the Data Warehouse (The fourth Edition)[M], Wiley, 2005.

[2] Mauricio H,Ward H,Felix P,et al.G.Clio:A schema mapping tool for information integration. Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks,2005:30-39

[3] Helena G.Data cleaning and transformation using the AJAX framework.Lecture Notes in Computer Science,2006:327-343

[4] Raman V,Hellerstein J M.Potter's wheel:An interactive data cleaning system.VLDB, 2001:381-390

[5] Y. L. You and X. Zhang. A reliable ETL Tactic and Framework Design for Data Warehouse. Computer Engineering and Application[J], 2005, 41(10): 172-174,229

[6] H. G. Chow, J. C. Chow, Y. Q. Peng. The General Data ETL Framework Design. The Computer Application[J]. 2006, 23(12):96-98

[7] Christian T, Pedersen T B. Research and realization of incremental ETL tool. Lecture Notes in Computer Science,2006:1-12

[8] Iqbal T, Daudpota N.XML based framework for ETL processes for relational databases. WSEAS Transactions on Information Science and Applications, 2006, 3(7): 1402-1406