2012 International Conference on Information and Knowledge Management (ICIKM 2012)
IPCSIT vol.45 (2012) © (2012) IACSIT Press, Singapore

The middleware processing layer for the batch data flows

Maros Zobok, Pavol Tanuska, Milan Strbo

Institute of Applied Informatics, Automation and Mathematics
Faculty of Materials Science and Technology
Slovak University of Technology

Abstract. This article is focused on the main issues during the data integration of heterogeneous information systems. The aim is to propose the data flow processing structure which will increase the data control, improve the procedural logic and create the effective way to trace data import failures and data discrepancies. The improved data flow processes will be implemented in the middleware import layer placed between the data source and data destination. The solution of the work offers the design concept of the data flows and the import object structure that allows the efficient processing of the data during the actual import and provides a multi-level control of processed data and their monitoring.

Keywords: data integration; data flow; process control; batch import; transactional database

1. Introduction

The information systems in most large manufacturing companies consist of many subsystems integrated together with a constant data exchange. In such environment we have to find how to deal with the most effective way of managing data transactions and system interaction to create the well working integrated system. It is not only about programming data flows itself, but in the case of transfer failures we have to have the possibility to trace these failures and to check and correct them. Successfully integrated enterprise system must be able to offer the functional data management on every enterprise level. The high-quality data management is characterized by the following features. It has the consistent process and data flows, ensures the availability of the data, reports errors and exceptions and provides analysis of the historical data.

The responsibility for data must be clearly defined. The targets for achieving the desired quality must be identifiable and measurable. Within the centralized approach the data quality is maintained and inspected upon the entry into the system. In the decentralized approach the data quality must be maintained in the data sources. In both approaches there must be developed control mechanisms that can detect errors quickly enough and at the same time allow them to eliminate it effectively. Such activities should also be measured and reported. Most companies usually monitor the quantity of defects, their type and the correction speed. If the company fails to provide quality data entries, then they will never be able to provide correct data outputs and data management.[1]

2. The main data flow integration deficits

Studies have shown data integration to be the number one headache of most customer-focused projects. The truth is that most project leads tackling customer management initiatives knew deep down that data integration was hard, so they saved it for last. In fact, for most enterprise-wide technology initiatives, data was an afterthought. The state of most companies' information was downright perilous, with data waiting in the tall grass of the corporate infrastructure, and no one wanting to get close enough to it to see what it really looked like because they knew it would bite. [2]

2.1. The lack of control and monitoring processes

Since the data transmission is not only the simple transferring information among the systems, but users must also be able to process the transferred data, so that the process of implemented import is as efficient as possible. In addition to a simple transfer of data it is essential that the user has the control over a single data processing and data calculations. Each processed record should be validated in the sense not to damage the master data in the target managerial ERP system. The records that are rejected by the import process should be marked for the future control or data corrections. Finally, records must be archived and if needed, appropriate data file must remain accessible and usable for the subsequent information tracing.

Typical issue of such systems is the lack of feedback control and un-effective log functionality. It is always handy to get a feedback on imported information, to know the status of processing and to get this information easily accessible. This is often forgotten in the planning phase. Once we implement it into the system then the operations will be grateful. We often struggle with data import failures and then we have to search for information about import status of single records or we process data and we can't find quickly in the logs what was the failure reason.

Sometimes we use applications with the large scale of functionality, but the suitable log functionality is missing. The processed data are usually logged in one single table from where it is quite hard to get required information but even worse the logs are stored in one single flat file. If we imagine thousands of daily imported records this is really the significant issue. Our effort to process control will constantly break down without proper data loads and reliable data integration. [3]

2.2. The middleware application layer

To eliminate deficiencies in the data processing during the import we will propose a middleware application layer between the data source and processing of the source data by the integration application. From this staging layer the data will be moved to the integration application and then finally loaded to the destination system. This middleware application layer will meet the control functionality as well as serve to store the results of imports. It will have a precise definition of the structure which will contain a set of procedures that will sequentially process the imported data to get the import into the target application smoothly and efficiently.

The possibility of management and control of data flows will be automated. The data will be checked prior to import, the results of the inspection will be evaluated. Then they will be processed according to the requirements of target applications and the result will be reported by sending the required notification message. The user will get full control over all data imports, speed up the processing and identification of failed records, damaged or rejected imports. The failure identification will be traced to the level of individual records, entities and their attributes.

3. The proposal of the effective batch data flow process

3.1. The import object structure

In this case the importing structure will act as an intermediate layer of data import implemented in a temporary data storage. In this environment there will be implemented all the features necessary for evaluating of the data flow processing effectiveness. The data here are preprocessed, checked, ready for import and eventually archived. According to these functions there will be prepared the total final structure of the objects.

The basic structure consists of the five main objects: Import, Master, Result, Log and Supplement (see Fig. 1). For each batch data import it is necessary to create a facility-specific logical structure with its own physical data storage. There are defined the five basic objects that can be further extended, if necessary.

Object Import: The object contains a mirror view of the source data and its structure is consistent with the view of the source database from which data were exported into a data file.

Object Master: This object contains an overall image of the target data to be updated by the batch import. Data are exported from the target system at the time of the data flow. Moreover, it may also be used for data processing during the data import.

Object Result: After executing the necessary data calculations and modification, data are transferred to the resulting object from which the data transfer will be accomplished. The data will be divided according to the symptoms, whether they are intended to insert a new record into the target data storage, update or delete the existing data record in the target structure.

Object Log: After processing and importing the data all the records are transferred into the Log object, including information about the status of the import, time and any error messages.

Object Supplement: The object is used for supporting the data modifications and contains all the supporting data that are to be used for data processing. This procedure is used for the transformation or mapping tables.

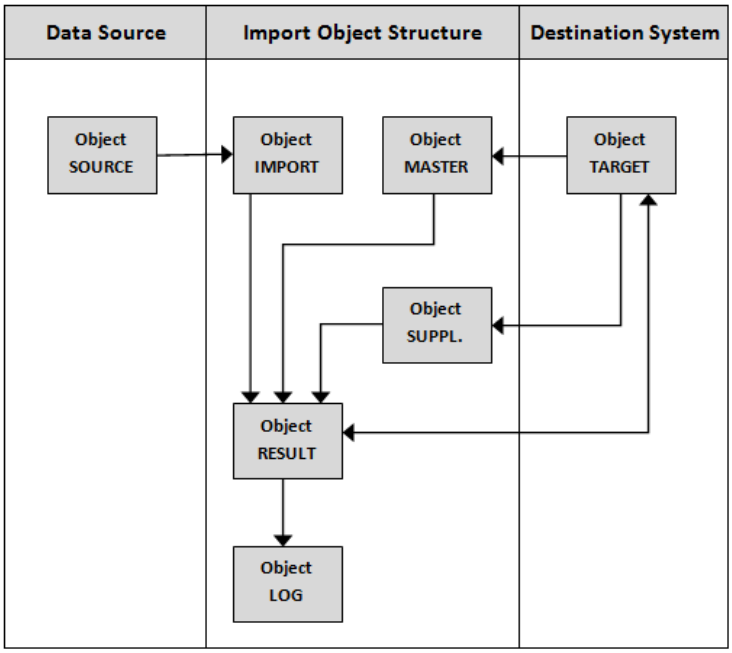

Fig. 1: Conceptual model of the importing object structure.

Intermediate results, final results and auxiliary information such as the time stamp or symptoms of the data processing operations can be recorded and maintained directly with the processed data. For the purpose of recording the progress and results of the data we must assign attributes where we will store the necessary values. [4]

3.2. The application data flow processing

The data flow processing is implemented by the set of SQL procedures. There is placed an application semaphore at the beginning of the process. As the batch data import is automated, we must ensure that we will avoid a situation where the new batch import starts but the previous one is not finished yet. That would cause the tables being populated with data in the wrong order and the whole process would fail. Then a source file is inspected after checking the semaphore. We have to verify the existence of the file in the import directory and the date and time of the file creation. If the source data are loaded directly from the source repository view we can check the database connection in this step. If the file does not exist in the designated directory or does not meet the conditions for the import, the system will generate the notification warning message.

After the source file passed the control, the procedure will start the truncation of import stateless *Import* and *Master* objects and loading the fresh values. The source data values are loaded into the *Import* object structure. The objects *Master* and *Supplement* are populated for the need of additional data processing. Afterwards the data will begin to be processed. The data processing depends on the specific demands

for each data flow. Now the data can be transformed into the required formats, calculated to the required values, aggregated, cleaned etc.

Then there occurs the result check after processing. In this step we can set the qualitative and quantitative criteria to start the import itself. The qualitative criteria may check the accuracy or completeness of the fulfilling of the specified data record attributes. The quantitative criteria represent the boundary conditions for an amount of imported records in the system. If the batch contains less data records than the minimum threshold condition, it may indicate that the data file for the import is incomplete or corrupted. If the system records a higher amount of entries than the maximum boundary condition, it indicates that there could occur unexpected changes in the source system. Both cases generate the notification messages and the import processing must stop in this step. The faulty data entries demand additional examination. In case there are no changes identified during the batch processing between the source and target attributes, then such a record will be marked as ignored in the import. This will speed up the actual data import because it does not overwrite the data with the same value.

If we can define criteria precisely, then this control may be fully automatic, but in some cases, human intervention is inevitable. Anyway sometimes it is needed to contact technicians responsible for the source data. It is necessary to consult the results of the inspection directly with them. For example, if they made major system changes, then the quantities of the changed attributes would increase unexpectedly. But such increased amount will also be evaluated as valid.

The data ready for the import are transferred to object *Result*. In the object *Result* we also have to store necessary additional attributes of the record like the status of the import, timestamp and an error message in the case of the failure. If the control criteria are met, the import procedure will activate the integration application. Then the records will be transferred to the target ERP system. After the transfer is finished, then all the data are archived in the object *Log*.

3.3. The supplemental processes

The basic data processing can be supplemented by any additional subprocesses which will extend the functionality of the data load. As we can receive different namings during the data integration for the same entity, the subprocess of the data value transformation must be implemented (see Fig. 2). The terminology may differ due to the usual practice on the individual workplaces or due to manual data entry errors. If we collect such data we need to consolidate these different names per entity to a single name which is used in the central data repository of the target ERP system.

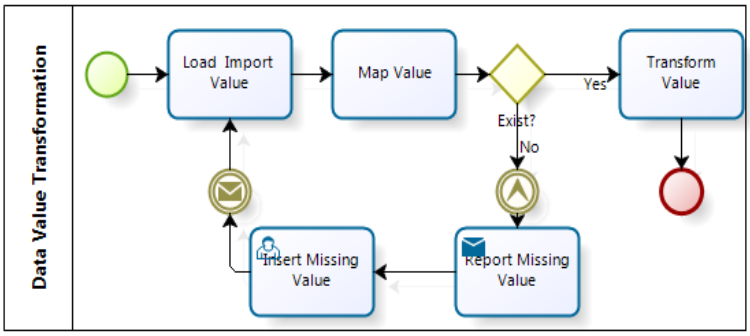

Fig. 2: The subprocess of the data value transformation.

For this purpose we have to create a mapping or transformation table, which is part of the object *Supplement*. It contains the list of all the names of the entities that differ in data collection. Then these different names are aggregated into a consolidated name. Afterwards the original data value is translated and imported into the system. If, however, we load a value that cannot be matched in the mapping table, we need to report this value and initiate the assignment of record pertaining to the naming in the target system. Finally this data pair is added to the mapping table.

The subprocess of verifying the results of data processing is used for the automatic control of processed source data before importing them into the target system (see Fig. 3). Such control is implemented for the protection of the target data from their damage during an unexpected import of the wrong values.

First of all the user defines conditions that should meet the processed data. Conditions may relate to the total amount of data such as maximum or minimum amount of updated, inserted or deleted data or anticipated changes in values and quantities of attributes. The checking amounts for the data results can be disaggregated to lower levels down to the data attributes changes.

If the system identifies smaller amount of imported data than is the expected value for the import into the system, it may mean that the source file was not loaded completely. On the contrary, if the resulting value exceeds the maximum specified limit, the system will indicate the unexpected changes in the source system from which source data were exported. In this case, the import process stops and generates a notification massage. After confirmation of the results the processing can be executed and data are transferred to the target data repository.

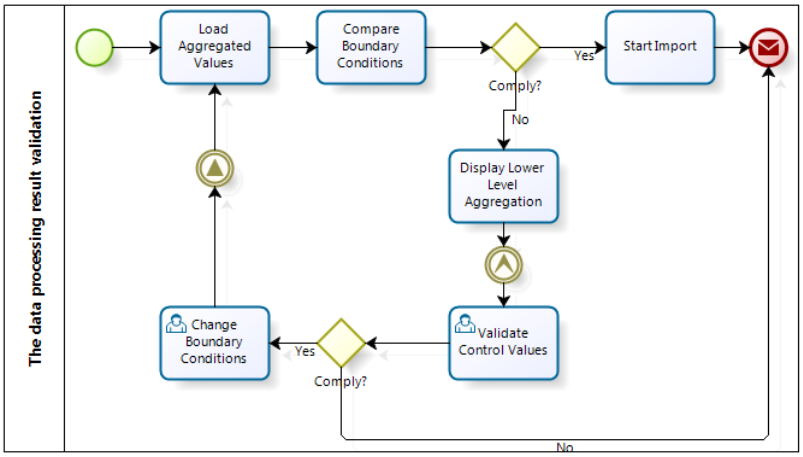

Fig. 3: The subprocess of the process result validation.

4. Conclusion

The main objective was to ensure the data consistency of the integrated system. Only in this case the management system can generate outputs for the effective management. This implemented solution offers a superior option to maintain the integrity of data flows and provides system administrators access to detailed information about the progress and the results of the data imports. The structure of the system keeps detailed information about the data flows which are used to identify and correct import errors. The system does not only detect and display the error but also the cause. The automated control mechanism incorporated in the solution prevents the damage of the data and referential integrity in the target system.

5. References

[1] A. Coldrick, J. Clement, J. Sari, "Manufacturing Data Structures: Building Foundations for Excellence with Bills of Materials and Process Information", Wight (Oliver) Publications Inc., 1992, 276 pages, ISBN: 0939246279.

[2] J. Dyché and E. Levy, "Customer Data Integration: Reaching a Single Version of the Truth", John Wiley & Sons, 2006, 320 pages, ISBN: 9780471916970.

[3] M. Zobok, P. Tanuska, P. Vazan, "The Integration of Processes within IT Resource Control," The 3rd International Conference on Advanced Computer Theory and Engineering ICACTE 2010, Chengdu, China, p. 222-226, ISBN: 978-1-4244-6540-8.

[4] M. Zobok, P. Tanuska, "The integration processes for the effective dataflow control and monitoring", In The International Conference on Computer Science and Service System CSSS 2012, Nanjing, China, ISBN: 978-1-4673-0719-2 , in press.

[5] M. Zobok, P. Tanuska, "The Preprocessing of Data Imports within the Integration of Heterogeneous Systems", In International Conference on Information and Computer Applications, ICICA 2011, Dubai, UAE, p. 78-82, ISBN 978-1-4244-9503-0.