A Multi-Stage Approach to Asymmetric Legacy Information Integration

Y. Tang[1], J.B. Zhang[1], C.H. Tan[1], M.M. Wong[1] and T.J. Ng[1]

[1] Singapore Institute of Manufacturing Technology, 71 Nanyang Drive, Singapore 638075

Abstract. Ensuring data integrity in the process of integration with heterogeneous legacy systems has proven to be a very challenging task because of the "asymmetric" nature of the integration. This paper first identifies and describes the major issues in the integration process with legacy information systems. A generic five-stage integration approach is then proposed to address the issues in a systematic manner. The proposed data manipulation and infusion methodologies together with the schemata and how they can help isolate the heterogeneity of the integrated systems and provide high transparency to the applications are then discussed. In addition, we shall discuss how event-driven active mechanisms can be applied to ensure a high level of integrity, flexibility, availability and reliability for the asymmetric integration with heterogeneous legacy systems. The proposed framework has been validated in an industrial project, and has enabled the seamless integration and continuous information flow between a new corporate information system and the legacy production and material handling systems.

1 Introduction

The advent of computers and the wide acceptance and implementation of information technology (IT) have brought about tremendous benefits to many organisations especially in terms of data storage, information processing, distribution and controlled access. For most organisations, various IT systems have been implemented at different time periods. As a result of this, data and information of an organisation are scattered around in many isolated and disparate information systems that are developed at different times, with different technologies and on different platforms. Driven by increasing intensified competition, today's manufacturing companies are forced to find ways to integrate their internal information resources in a seamless manner. By doing so, their competitiveness will be enhanced by way of improved efficiency and reliability in information acquisition, processing and dissemination. This will lead to better and more effective decision support capabilities, and allow collaboration amongst various departments of the organisation, as well as supply chain partners.

However, "seamless integration" has so far been more of a wish than a realization, due to the difficulties that are faced in dealing with the heterogeneity, obsolete tech-

Tang Y., B. Zhang J., H. Tan C., M. Wong M. and J. Ng T. (2004).
A Multi-Stage Approach to Asymmetric Legacy Information Integration.
In *Proceedings of the 2nd International Workshop on Verification and Validation of Enterprise Information Systems*, pages 50-59
DOI: 10.5220/0002671100500059
Copyright © SciTePress

nology and semantic discrepancies inherent in legacy information systems. The "heterogeneity" of information systems can be classified into the following levels [7]:

- Platform level (hardware, operating system and network protocol)
- Data management system level (data query languages such as SQL, data implementation models such as object, relational, hierarchical and network database)
- Location level (where the data resides)
- Semantic level (multiple, replicated and conflicting representations of similar facts)

To some extent, technologies such as SQL, ODBC, JDBC and CORBA have helped to resolve issues that are related to platform and data management system heterogeneities. However, despite much effort spent by the scientific community, semantic discrepancy is still an open and largely unsolved problem [12].

Data extraction, transformation and loading (ETL) for data warehouse is software that can be used for the extraction of data from several sources, cleansing, customization and insertion into a data warehouse [14]. Although ETL tools can be used with legacy information systems, their focus is on historical information rather than current (operational) data. In this paper, data integration for dynamic environments will be discussed. The focus is on integration on demand rather than integration in advance to allow for "continuous flow" of information between dynamic systems such as production control, MES, and so on.

A Federated Database System (FDBS) is a collection of autonomous database systems that cooperate to provide a combined view of individual data stores [10]. We feel that the conventional approaches of FDBS are inadequate for the integration with legacy information systems since it is difficult to make changes in all legacy systems so as to extract data from local stores and transform data according to the "federation requirement". The typical integration processes with legacy information systems will be asymmetric, i.e., the majority of data transformation jobs are carried out in the new environment rather than distributed throughout the individual systems.

Hence, in this paper, we are proposing an architectural framework and the techniques to address issues relating to the integration of legacy information systems with new information systems. The ultimate challenge is to ensure data availability, data integrity, and maximum system flexibility for the asymmetric integration with legacy information systems. Typical constraints of asymmetric integration include:

- Changes are not allowed to be made to the legacy information system due to high integration costs, unavailability of the human expertise, or the mission critical nature of such systems does not permit disturbances to be introduced;
- Further additional loading of the legacy information system is not allowed because of the limited system capacity;
- Unique identifiers for data entities to be shared cannot be provided by legacy system, even though such unique identification is required by the receiving systems.

The objective of this work is therefore to design an integration framework that will allow reconfigurable event-driven and on-demand data exchange between the integrated systems, including legacy information systems. Related work such as ETL for data warehouse and Federated Database System are reviewed in Section 2. In Section 3, we will discuss the major challenges in legacy information system integration and propose a five-stage integration approach. Active mechanisms to ensure data integrity

are present in Section 4. Section 5 describes industrial implementations, and finally, Section 6 provides the conclusion.

2 Related Work

In this section, we will review some related work in the fields of databases and information systems integration, and discuss their suitability to be used in legacy information systems integration.

2.1 Data Extraction, Transformation and Loading (ETL) for Data Warehouse

Data extraction, transformation and loading (ETL) are the set of functions, which reshape relevant data from source systems into useful information that will be stored in the data warehouse [6]. It includes the following major functions:

- **Data Extraction**. This function deals with numerous data sources in diverse data formats by capturing data from various sources such as transaction logs, database triggers, application source codes, and by comparing the timestamps of records.
- **Data Transformation**. This function converts raw data into the format that is usable in the data warehouse. It often involves the following steps: selection, splitting/joining, conversion, summarization and enrichment.
- **Data Loading**. This function involves populating the data warehouse tables for the very first time and applying ongoing changes in a periodic manner.

ETL tools have been extensively studied to supply data warehouse with clean data (e.g., [1], [6], [8] and [14]). We have evaluated and found that some ETL techniques are suitable for the proposed data integration framework. Since the main purpose of data warehousing is to perform data mining and decision support, the standard ETL tools tend to focus on historical information rather than current (operational) data. In contrast, we focus more on data integration in dynamic environments (such as the manufacturing information system model that will be studied in section 3), and this requires integration on demand, rather than integration in advance.

2.2 Federated Database System

The Federated Database System (FDBS) was introduced to provide a uniform access to information that is physically distributed [10]. A FDBS is defined as a collection of autonomous database systems that cooperate to provide a combined view of individual data stores. Various models and methodologies for FDBS have been proposed by numerous researchers [2], [9] and [12]. In general, the architecture is made up of a few layers, including, for example, Local Schema, Component Schema, Export Schema, Federated Schema, and External Schema. Each layer presents an integrated view of the concepts that characterize the underlying layer, and taken together, supports the distribution, heterogeneity and autonomous features of the FDBS.

However, the aforementioned FDBS architecture requires individual component databases to extract data from their local stores and transform the extracted data according to the "federation requirement". While this approach works for newly established database systems, it faces great difficulties when dealing with legacy information systems that were developed years or decades ago that do not support the latest database technologies, not even support the de-facto database language, SQL.

In most cases, the legacy information systems in an enterprise may not be replaceable or even upgradeable, due to the high costs and organizational risks, or technological limitations that are associated with such reengineering [11]. They are usually mission-critical and not modifiable without a thorough study and understanding of the original design. Many of these systems are often running near their full capacities and the hardware platforms are no longer upgradeable or even supported by the original vendors. As a result, the integration processes between newly developed systems and legacy systems are normally asymmetric. In other words, the majority of data transformation jobs have to be carried out in the new environment rather than distributed in all component systems including legacy databases. As such, the conventional approaches of federated database system are inadequate for the integration with legacy information systems. Hence, an architectural framework that is suitable for legacy information systems integration is needed.

3 A Generic Framework for Legacy Information Integration

3.1 Challenges in Legacy Information Systems Integration

In this section, a typical scenario is presented to highlight the major integration issues that this paper aims to address. The information integration in this example comprised of two independent heterogeneous databases within a manufacturing company; one

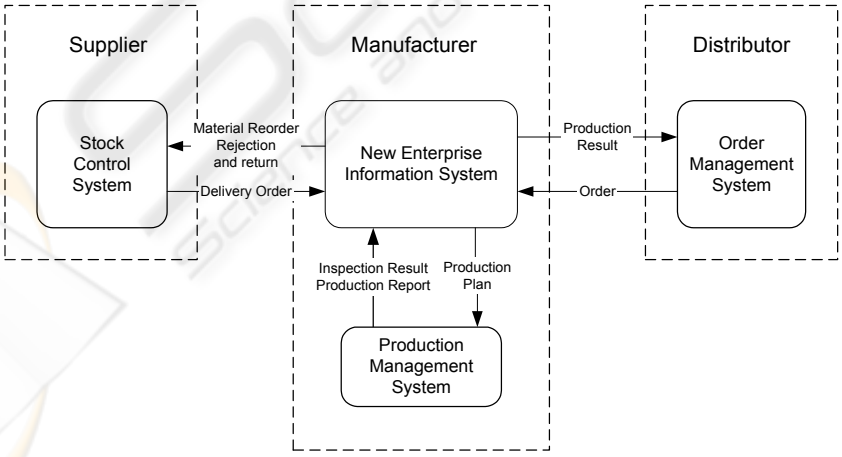

Fig. 1. Inter-Organizational data flow

being a legacy Production Management System (PMS) for real-time production control and quality management, and the other being a new Enterprise Information System (EIS) that supports production planning, inventory management and management report. The new EIS also acts as the communication gateway with the information systems of supply chain partners, transferring material usage information to the upstream material supplier and finished production order information to the downstream distributor. The flow of data amongst these systems is shown in Fig 1.

The Production Management System was developed many years ago together with the production lines, and is already running near its full system capacity. Before the new EIS is introduced, there were several isolated computer systems in the manufacturing company for planning, reporting and so on. The new EIS needs to integrate seamlessly with these disparate information systems of the organization as well as those of the supply chain partners, to reduce the manual data entries and re-entries which are labor-intensive and error-prone. The most problematic part is interfacing with the legacy PMS, which mainly transfers production data from the PMS to the EIS. The key issues to be addressed in this example include:

- The implementation has to be carried out in the new EIS as much as possible in order to minimize disruptions in the legacy information system.
- Both periodic batch processing and real-time transaction-based data exchange has to be supported.
- The PMS maintains the production data on a daily basis and transfers all daily production results to the EIS several times a day, without differentiating data that have already been transferred by previous transfer sessions.
- As a data repository for the PMS but also an entity in the supply chain, the EIS has to differentiate data entities transferred from the PMS in different batches, as portions already included in the management reports or passed to supply chain partners cannot be altered any more.

This example represents the situation that is commonly found in many other industry sectors in addition to manufacturing. It is therefore important to establish a generic approach that can be applied to solve a class of legacy information systems integration problems that are typified in Fig 1.

3.2 The Architecture Framework

The proposed data integration framework for legacy information systems integration, given in Fig 2, adopts a five-stage integration approach with the following key processes: Extraction, Filtration, Aggregation, Reconciliation and Infusion. The corresponding data repositories that support these processes are: Interface Schema, Local Schema, Consolidated Schema, Integrated Schema and Application Schema. Each stage presents an integrated or processed view of the data that has been manipulated in the previous processes. The characteristics of each stage of the integration process and the corresponding schema are discussed in the following sub-sections.

3.2.1 Extraction Process with Interface Schema

The first stage of the integration process is the Extraction process. The Extraction process operates on a set of interface tables and protocols that serves as the direct communication interface with the legacy systems. Taking the system heterogeneities into consideration, the data extraction process may be performed in many forms, ranging from the tightly coupled data exchange technology to loosely coupled data batch conversion, import and processing. If the source system is a RDBMS that supports transaction logs or database triggers, or if the source application can be modified to capture the new data in real-time, the data extraction is immediate. However, features such as transaction logs or database triggers are normally not available in legacy systems and modification of the legacy programs may not be feasible. In this case, deferred data extraction will be the only option, and the interface schema will provide a snapshot of the data in the legacy system, transferred by a periodically batch process. Most ETL tools for data warehouse also contain Data Extraction stage.

3.2.2 Filtration Process with Local Schema

Owing to different functionalities of the EIS and PMS, raw data in the interface schema that is obtained from the legacy system, needs to be filtered before it can be used by the EIS, and the Local Schema only contains data that is of interest to the EIS. Logically, the data filtration processes should reside in the data source system, as it has far better control over how data can be filtered. For example in a FDBS, a filtering processor is usually applied between the Component Schema and Export Schema of each component DBS. In the case of legacy integration however, this may not be practical as it is critical to minimize the disruptions, uncertainties and risks that any

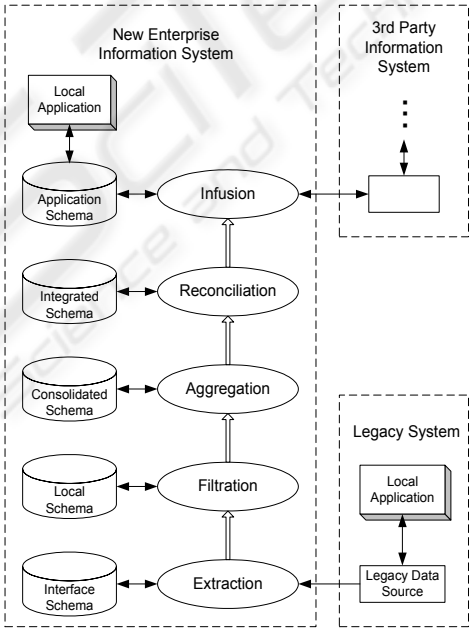

Fig. 2. The five-step integration approach

modification may introduce to the legacy system. Consequently, the integration tasks have to be performed by the new EIS.

3.2.3 Aggregation Process with Consolidated Schema

Different systems manipulate production related information at dissimilar levels of detail. For the same attribute of a data entity, the EIS and PMS may take on different values. For example, WIP statuses in the PMS like "Operation Start", "QA Testing", and "Reworking" could all mean "In Operation" for the EIS. In the Aggregation stage, a combination process will convert existing attributes to a common attribute definition by using the system-mapping dictionary and then merge all entities that have the same common attribute definition. It provides attribute definition transparency for the upper layers by isolating the semantic heterogeneities of the integrated systems.

Solving semantic incompatibilities for database schema integration has been studied in the database realm for a long time [3], [11], [12]. To establish relations with the entities of a legacy database is extremely difficult, because the semantics of legacy system could be in a default or implicit form. Apart from collecting semantic knowledge from available schemata and data-dictionaries, the integration designer also needs to collect information from the legacy system administrator, user and historical data. The acquisition and representation processes of semantic knowledge for legacy system are highly domain-specific and cannot be performed automatically.

3.2.4 Reconciliation Process with Integrated Schema

A major challenge that the legacy system integration architect has to face is how to ensure data integrity, especially if the legacy system cannot distinguish whether or not the data has been transferred in earlier updating sessions, and therefore always provide the entire set of data. The situation will be further compounded when the legacy system cannot even provide unique identifiers for the data entities that are transferred.

As an example, a PMS may accumulate records of rejected raw material that are generated by the quality control stations, and transfer to an EIS several times a day. There could be multiple entries for the same type of materials, by the same quality control station, but at different times. The timestamp of the data may be available but it will change when the materials are reworked or retested. The EIS needs to differentiate the data received from the PMS in different updating sessions, as the previously received data has already been communicated to the material supplier. Data capture that is based on date and time stamp using the normal data extraction technique of ETL approach is not feasible in this environment.

In order to satisfy the foregoing requirement, the Reconciliation process is introduced. An accompanying Integrated Schema serves as a repository for data entities that have already been previously transferred and assigned with unique identification values. The summarized data of the entries in the Consolidated Schema, which is now distinguishable, will be compared with existing records in the Integrated Schema. Only net quantities that represent the "new" records will be appended into the Integrated Schema and assigned with unique identifications. As a result, the Reconcilia-

tion stage resolves system heterogeneities that are caused by differences in the timing and frequency of data exchange.

3.2.5 Infusion Process with Application Schema

The Application Schema is the working domain for the local application of EIS. The Infusion process provides the last stage abstraction and transformation of data from the PMS. It prepares data in accordance to the standard required by the EIS local application, and merges them with other existing data in the Application Schema.

In certain situations, the local application is allowed to enter new records or manipulate existing records transferred from the legacy system, in case the legacy system or the integration interface has been down for a period of time. However, this kind of manipulation brings about another consideration to the design of the Infusion process. When the key value of incoming record from the legacy system matches with the key of an existing record, either a constructive merge or a destructive merge could be used, and the selection will have to be based on the nature of the individual tables.

3.3 Flexibility of the Proposed Framework

The generic five-stage integration approach presented in this paper addresses the general issues that are faced in most legacy information systems integration situations. However, it is possible that in a specific implementation, not all stages of the integration process are needed. Instead, variations to the integration processes can be made based on the following guidelines:

- In cases where the data source systems are able to perform data filtration according to the requirements of the target system, the Filtration process and the Local Schema can be considered as redundant.
- If a common data-mapping dictionary exists and is available to all integrating parties, and the data source systems are able to perform the necessary consolidation, there is no need for the Aggregation process and Consolidated Schema.
- When the data exchange is in real-time and transaction-based, and all transferred data entities have unique identities in all integrated systems, the Reconciliation process and the corresponding Integrated Schema are not required.

4 Active Mechanisms to Ensure Data Integrity

Ensuring data integrity is critically essential for multi-source information integration. Due to the technological limitations of legacy systems, many data integrity constraints have been embedded in the application code or may have to be enforced in manual operation procedures. This will give rise to the possibility of potential data integrity violations in legacy information systems integration.

Event-driven active mechanisms are devised in this work to overcome the violation of data integrity constraints. Instead of stopping the data processing or simply reject-

ing the data when a violation occurs, active mechanisms allow recovery procedures to be invoked with minimal interference to the business process. Such active behaviour can be generally expressed by the Event-Condition-Action rules (or ECA-rules) in active database systems ([4], [5] and [13]). As an example, a typical active rule to recover from the above mentioned constraint violation could be defined as:

rule *Uniqueness_Constraint_Recovery*
on event *which potentially violates the uniqueness constraint*
if condition *the data entity does not have an Aggregation process with Consolidated Schema*
do action *reroute the process and include the necessary data preparation for the next stage*

Duplicated records will be consolidated by the re-introduced Aggregation process and stored in the Consolidated Schema. The Reconciliation process will be carried out on the assumption that there is no occurrence of exception yet. Supported by active rule mechanisms, the integration framework is configurable and adaptable in real-time.

5 Industrial Applications

The proposed five-stage integration approach together with active mechanisms provides a flexible integration framework that will ensure a high level of data integrity and availability when dealing with problematic integration of legacy systems. The methodology has been validated and implemented in an industrial project, and has enabled the seamless integration and continuous information flow between a new corporate information system and the legacy production and material handling systems. The successful implementation has eliminated a tremendous amount of manual data entry activities that are tedious and error-prone. The commissioned system has been proven to be reliable and robust, while guaranteeing data integrity.

Integration with legacy systems in a heterogeneous environment is an unavoidable task for many information system developments and seamless integration with guaranteed data integrity, availability and reliability is always crucial. The methodology described in this paper can be applied to a wide range of industries such as manufacturing, logistics and the service industries.

6 Conclusion

This paper has identified and analyzed critical issues in information infusion from multiple data sources, especially with regard to legacy information systems. The limitations of conventional approaches such as ETL for data warehouse and Federated Database System (FDBS) architecture have been discussed. A generic five-stage integration framework has been proposed to resolve problems that are commonly faced in legacy information systems integration. It can be used as a generic reference architecture for the design and implementation of legacy information systems integration. The five-stage approach has been applied to an industrial application involving

the integration of several legacy information systems, including a realtime control system for manufacturing material handling and inventory management.

The data manipulation processes together with the schemata, isolate the heterogeneities of the integrated systems and maintain the integrity of data. Event-driven active mechanisms for integrity violation recovery provide a measure of flexibility to the generic framework. In actual implementations, the number of integration processes and the corresponding schema layers could be varied from system to system, or even from one type of data to another within the same system. A high level of integrity, flexibility, availability and reliability is ensured for the asymmetric integration with heterogeneous legacy information systems.

References

1. Cui, Y., Widom, J.: Lineage Tracing for General Data Warehouse Transformations. The Int. J. on Very Large DBs, Vol. 12-1, (2003) 41-58.
2. Kaji, I., Kato, S., Mori, K.: Autonomous Data Consistency for Cooperative Applications to Fairly Ally the Data in Heterogeneous Systems. Proc. Int. Workshop on Autonomous Decentralized Syst. (2000) 196–204.
3. Lukovic, I., Mogin, P.: An Approach to Relational Database Schema Integration. IEEE Int. Conf. on Syst., Man, and Cybernetics, Vol. 4, (1996) 3210-3215.
4. Mylopoulos, J., Gal, A., Kontogiannis, K., Stanley, M.: A Generic Integration Architecture for Cooperative Information Systems. Proc. First IFCIS Int. Conf. on Cooperative Inf. Syst., (1996) 208 -217.
5. Paton, N. and O. Diaz: Active Database Systems. ACM Comput. Surveys, Vol. 31-1, (1999) 63-103.
6. Paulraj Ponniah: Data Extraction, Transformation and Loading, in Data Warehousing Fundamentals. Wiley Publication, (2001) 257-290.
7. Peter McBrien and Alexandra Poulovassilis: Distributed Databases, in Advanced Database Technology and Design. Mario Piattini. London, (2000) 291-327.
8. Rifaieh, R., Benharkat, N. A.: Query-Based Data Warehousing Tool. Proc. the fifth ACM Int. workshop on Data Warehousing and OLAP, (2002) 35-42.
9. Roantree, M., Keane, J., Murphy, J.: A Three-Layer Model for Schema Management in Federated DBs. Proc. 13th Hawaii Int. Conf. on Syst. Sci., (1997) 44-53.
10. Sheth, A. and Larson, J.: Federated Database Systems for Managing Distributed Heterogeneous and Autonomous Databases, ACM Comput. Surveys, Vol. 22-3, (1990) 183-236.
11. Song, W.W.: Integration Issues in Information System Reengineering. Proc. 20th Int. Compt. Software and Applications Conf., (1996) 328 -335.
12. Thiran, P., Hainaut, J.-L., Bodart, S., Deflorenne, A., Hick, J.-M.: Interoperation of Independent, Heterogeneous and Distributed Databases. Proceedings. 3rd IFCIS Int. Conf. on Cooperative Inf. Syst., (1998) 54 -63.
13. Turker, C., Conrad, S.: Towards Maintaining Integrity of Federated Databases. Proc. the 3rd Basque Int. Workshop on Inf. Technol., BIWIT '97, (1997) 93-100.
14. Vassiliadis, P., Simitsis, A.: Conceptual Modeling for ETL Processes. Proc. the 5th ACM Int. workshop on Data Warehousing and OLAP, (2002) 14-21.