Framework for ERP Integration with Legacy System

Bethel Endrias
HiLCoE, Software Engineering Programme, Ethiopia
Ethiopian Trading Enterprise With Cimac Inc.
letebendrias@gmail.com

Mesfin Kifle
HiLCoE, Ethiopia
Department of Computer Science, Addis Ababa University, Ethiopia
kiflemestir95@gmail.com

Abstract

Enterprise Resource Planning (ERP) is an industry term for the broad set of activities that helps an organization manage its business. ERP system use for supporting business processes is growing at organizational level. Nowadays, ERP systems are being implemented in numerous organizations as part of their business development strategies. However, organizations have not achieved many of the anticipated benefits. Autonomous and heterogeneous applications co-exist in companies with ERP systems and integration problems with legacy systems have not been addressed entirely. In ERP integration with legacy systems, the variation of application background and data transaction methods of ERP systems from legacy systems are the main problems creating difficulty on ERP implementation. This paper is aimed at assessing the current experience of ERP integration in larger organizations by identifying the challenges of ERP systems and legacy systems integration. In this work, information was gathered by interviewing ERP users, document analysis, and observation of project implementations. The collected data were analyzed using qualitative methods.

This paper proposes an effective business data incorporation framework among heterogeneous legacy systems with ERP. The proposed framework will allow the legacy systems or other enterprise applications to communicate or make smooth data flow with the existing ERP system. It is developed based on the data collected and interview results. The framework is evaluated using focus group discussion.

Keywords: ERP System; ERP Architecture; Legacy System; ERP Integration

1. Introduction

Application integration is a strategic approach to bind information systems. Since the time ERP systems were developed, companies were struggling to integrate ERP systems with legacy systems. ERP systems amplified the need for integration, as existing systems have to be incorporated with ERP applications. [1].

ERP systems automate core business activities without affecting underlying business structures and processes. As a result, a number of disparate applications often coexist with ERP systems. Integrating ERP with legacy enterprise applications is a problematic area of ERP system integration. In addition to the ERP business information backbone, many companies have legacy systems for manufacturing, execution, data collection, quality management and process control, warehouse management, transportation management and/or other “outside” systems that could benefit from integration with the ERP backbone.

The problem of integrating ERP applications with legacy systems is as old as the ERP industry itself. Typically, organizations are still working to improve integration between the ERP modules and legacy system long after the implementation. The main problem faced implementing an ERP system, besides unrealistic implementation timelines and expectations, is integration with legacy systems.

Mostly of the time incompatibility occurs when the application type changes, due to the database type, data flow changes, and the fact that most of the frameworks proposed to solve such problems are dependent on specific type of legacy system or ERP application, e.g., the solution planned for SAP ERP might not work on MSDN ERP system or the framework could be only to make report transaction rather than database communication and effective data flow. This kind of gap between the existing frameworks makes the integration of ERP with legacy systems bring extra cost like time, money, human resource, etc. Currently implementations that are going on in organizations use different methods to integrate ERP with the existing system that is because different ERP systems integrate with different governmental and non-governmental organizations that use numerous systems for their data and information handling.

The main objective of this paper is to develop a framework to integrate ERP systems with enterprise legacy systems. To solve the mentioned problems of ERP integration with legacy systems, we have to identify

- What the unique characters of ERP are,
- The challenges in practicing ERP integration with legacy systems, and
- How to integrate ERP with legacy system.

To answer these questions, we used systematic literature review to find out related researches and their findings. We also distributed a questionnaire and analyzed the responses. Interview was used to study the existing experience. We also visited and observed project implementations. The response from technical expertise from the selected organizations were almost the same, the organizations use inbuilt integration mechanism. Each of them develop unsuccessful frameworks and applications. Due to the incompetence with other legacy systems and large enterprise applications, they were not in use for much longer.

2. Related Work

Implementing an ERP system requires the services of many professionals that cost money. The major cost items are customization, integration, data conversion, data migration, testing and training [2]. To those involved in IT, systems integration has been one of the hottest catchwords in the past years. Researchers and practitioners differ in their definition of systems integration; but everyone agrees that systems integration is difficult and complicated, yet, not impossible [3].

There are two main goals that enterprises would like to achieve. One is to change their legacy applications and integrate them seamlessly so as to meet or even exceed the expectations of customers. The other is to upgrade the applications into an enterprise business intelligent system so as to make better decisions. There can be several solutions to achieve these goals: all-in-one solution, build a new enterprise business intelligent system which can fulfill all the functions of the legacy business applications, self-evolving and light level integration, or evolve the individual applications so that they can be easily integrated. A business intelligent system can also be developed in various ways: heavy-level integration, keep the legacy application unchanged, or build a new enterprise intelligent system to fetch all the business data from the individual applications. These solutions vary in risk, cost, schedule, long term benefit, precondition and real-time response [4].

The most common solution tried over the years has been developing interfaces between applications. These included interface programs and interface data stores. This worked fairly well when only a few interfaces were needed. But as the number of interfaces grow, the result was additional maintenance problem (inter-application spaghetti).

Even these days, some organizations try to integrate their applications through interfaces – but without much success. Others try to solve application problems using approaches like

- Replacing legacy systems with integrated COTS packages,
- Developing data and information store,
- Establishing central operational data stores,
- Implementing enterprise portals,
- Using middleware,
- Using XML, and

- Reengineering all applications to a single architecture.

All of these approaches have value and some will even provide at least temporary solution. However, unless they are business driven and model based, they are more likely to further compound the problem than providing a solution [5].

The other main problem with ERP integration is updating the system. It is not uncommon for the latest version of an application or ERP to break the integration so painfully established. This means more modifications and custom programming are required to keep things working. We may also find third party applications that were developed to integrate ERP packages with other applications. Whether with software from an ERP vendor or a third party, it is important to test the new environment and make sure the integration actually works as expected. It is also important to integrate these packages into the test environment as we proceed to install the ERP system. They need to be thoroughly tested just like the modules native to the ERP system [6].

Enterprise Application Integration (EAI) is the use of technologies and services across an enterprise to enable the integration of software applications and hardware systems. Many proprietary and open source projects provide EAI solution support. EAI is related to middleware technologies. However, EAI is challenged by differences in operating systems, database architectures and/or computer languages, as well as other situations where legacy systems are no longer supported by the original manufacturers [7].

3. The Proposed Solution

The proposed integration framework consists of ten modules as shown in Figure 1 to handle data transfer between the legacy system and ERP. The framework contains data type identifier, request identifier, data conflict management and others which make the receiving and sending of data easier.

The legacy system interface will link the data to the framework which will do the data transfer to the ERP system by filtering, identifying and generating generic data types which will flow into the master database of ERP and get processed. The output of the legacy system will be identified by the data log and converted into the received data type. This framework could work on every ERP application since SAOP and web service components are the main ERP components. These components work on the ERP system in the survey and as per the interview results.

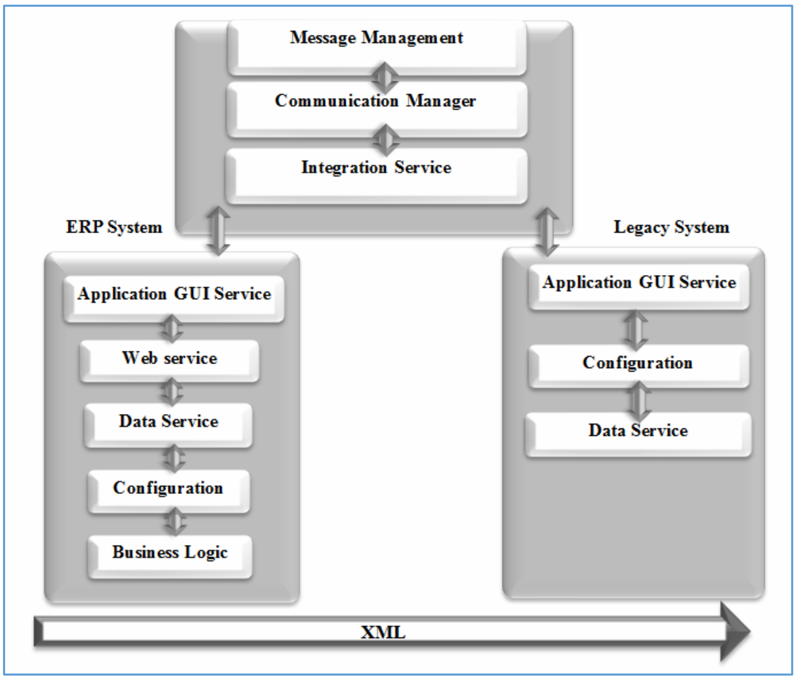

Figure 1: The Proposed Framework for ERP Integration with Legacy System

The integration service is a platform for building enterprise-level data integration and data transformations. It updates existing or legacy applications with stored database from ERP application. This application integration services also synchronize the business processes of ERP and legacy systems. Business rules, configurations and data transactions will be communicated on time through this service. This component is more interactive with the data that flows in and out of ERP and legacy systems. Integration service layer is not available in any ERP or legacy systems. However, the development environment is a way to build this service from the master application side which is ERP. This is where the whole integration idea takes place.

4. Discussion

The components incorporated in the framework are flexible enough to be used by open source legacy systems. In addition to that, the integration framework doesn't need any kind of interface designing and configuring like most of the integration frameworks do. It has a mechanism to respond to both systems in a way it receives the data by managing conflict and returning the needed output. As shown in Table 1, each of the framework components acts on the data that is entering into the system. Above all the components are presented to interact securely with the legacy system by converting and making secure data transmission. Since ERP system integration is capable of doing almost all enterprise work, when other applications need the integration, the main thing will be data exchange and the included components are capable of doing that. Therefore we believe that this framework will add additional input and further benefit on ERP integration with legacy systems.

The integration framework prepared in this study will solve this problem. Based on the survey, the implementation organization has given a testimony of variation of ERP system doesn't have huge gap of integration with legacy system. As mentioned before ERP system is central database based system. The data flow of ERP system integrated with legacy systems will be the bigger image of ERP system.

The security issue of the framework is mainly based on the two systems sharing the data. Basically ERP system has strong data security in its architecture. Thus the integration framework is relying on the architecture of ERP inbound and outbound data security besides its own method of clarifying inputs and outputs. Therefore, to search for practitioners' opinion on likely use or adoption of the framework in the time of two meetings would not be feasible. Providing sufficient knowledge of the proposed framework to draw meaningful and thoughtful comments would require investment that is bigger than this and which need time, resource and budget. The vendor companies liked the idea of having this framework and planned to incorporate it in their future work.

Table 1: Integration Service Framework

Service Name	*Task*
Inbound Service	This is where the incoming data are received.
Request Identifier	Identifies the type of request demanded from the data provider. The kind of data flow to the system will be identified by the data identifier and after that the type of process to happen on the data is identified.
Data Type Identifier	Data type validation answers such simple questions as "Is the string alphabetic?" and "Is the number numeric?". This process helps to feed the data log to return outbound service.
Data Integration Rules	Holds one or more systems with it. Multiple systems can insert data at the same time and if the systems want their data to be integrated and processed, it

Service Name	*Task*
	will be processed here.
Generate Generic Data Type	The incoming data might not be of the same data type as in the ERP database. This is because of the variation of third party systems to be integrated with ERP. Therefore, this method changes the incoming data of any type to a generic data type which will make it suitable for the ERP.
Reference Ontology	States a formal naming and definition of the data type format and metadata of the inbound and outbound flow of data.
Communication Structure	Controls the data information and data type that passes through the framework. The flow of data should incorporate structure in formalized way.
Data Conflict Management	Conflicts and errors are detected at the level of the row. The inbound service could provide data, which are not in acceptable format or the return type might not be acceptable. This component controls such issues.
Master Slave Management	This is where one of the system processes has unidirectional control over one or more other system processes. Most of the time, the ERP system takes the master communication since the integration is plugged on it.
Outbound Service	This is where the outgoing data are delivered to the legacy system.

One of the active interactive components of ERP with the integration framework is action Generator method which generates actions depending on data policy. These actions are implemented on the data passing through the interface. Web Service Generation Data Broker and Data Validation collect information about the data and pass it for circulation. Data validation is ensuring that a program operates on clean, correct and useful data. It checks for correctness, meaningfulness, and security of data that are input to the system. The rules may be implemented through the automated facilities of a data dictionary. Sending Service section distributes services processed in the database and external system.

In the same way Oracle E-Business Suite of applications provide comprehensive business solutions that can be implemented across a wide variety of industry functions including customer relationship management (CRM), project management, supply chain management (SCM), product life cycle management, financial management, and so on. Oracle application integration architecture is a catalog to pre-built integrations and an architecture that provides a foundation for integration between cross application business processes. Application integration architecture coupled with JD Edwards Enterprise One native SOA capabilities makes a solid integration offering that provides for the delivery of pre-built integrations in the form of process integration packs, such as the CRM Integration Pack for JDE Edwards Enterprise one that supports the lead to order process.

5. Conclusion

The main problem faced in implementing ERP systems, beside unrealistic implementation timelines and expectations, is integration with existing legacy systems, especially those which are made static in a company.

This work is about data transfer through legacy system and ERP system via components incorporated along with the framework which will be tagged along the ERP system and perform secure tunnel with the legacy system without any interruption. Therefore, no matter which package is chosen, the framework will function from the ERP side.

ERPs nowadays are sold in packages. It can be broken into smaller components and each component is purchased separately. This allows customization for each implementation. Choosing the correct package is essential. Reasons for improper package selection include lack of knowledge as well as lack of clarity about functional requirements. Sometimes packages are wrongly chosen because the characteristics of the package aren't well understood.

Organizational structure and ERP architecture highly affect ERP integration. Most companies have an environment of disparate legacy systems, applications, processes and data sources. Maintaining legacy systems is one of the challenges that enterprises are facing today. The commercial market provides a variety of solutions to this increasingly common problem. However, understanding the strengths and weaknesses of each technique is important to select the correct solution and the overall success of the effort.

The type of ERP doesn't affect the ability to integrate with legacy system, because the ERP systems are latest and up-to-date with almost the same system architecture and process. The size of an organization may influence the type of ERP to be implemented in a company. Therefore, companies should consult for the type of ERP they will implement. ERP integration with legacy systems is not only about data transaction. There are companies who want to integrate different ERP systems not only through data flow but also architecture wise. This depends on the infrastructure of the company and it could be a broad study area.

The framework proposed in this paper is generic and we hope it is possible to work on different types of integration areas, like sister companies who have different business architecture and use different applications and ERP systems (net system integration). The other open research area is the platform dependency for ERP integration that could be studied further. Security is also another issue in this area.

References

[1] Tommi Kahknen and Kari Smolander, "ERP Integration – A Systematic Mapping Study", Software Engineering and Information Management, Lappeenranta University of Technology, 2012.

[2] Rittik Ghosh, "A Comprehensive Study on ERP Failures Stressing on Reluctance to Change as a Cause of Failure", Journal of Marketing and Management, 2012.

[3] Zaitun Abu Bakar, "Success Factors to Systems Integration Implementation: More Technically Oriented than Human Related", Malaysian Journal of Computer Science, Vol. 14, No. 2, 2001.

[4] Bin Xu and Bin Xu, "A Data Secure Solution for Heterogeneous Enterprise Legacy Systems Integration", College of Computer Science & Information Engineering, Zhejiang Gongshang University, Hangzhou, China, Vol. 4, 2009.

[5] Alen Perkins, "Strategic Enterprise Application Integration", 2003.

[6] ERP Integration Challenges, http://it.toolbox.com/blogs/inside-erp/erp-integration-challenges-65516, Last accessed on February 24, 2015.

[7] Tariq Rahim Soomro and Abrar Hasnain Awan, "Challenges and Future of Enterprise Application Integration", International Journal of Computer Applications, Vol. 42, 2012.