Investigating Contemporary Software Engineering Challenges Involved in Integrating Disparate Software Systems

Nazeen Ansari[1], Dr. G.T. Thampi[2]

[1]Research Scholar, Thadomal Shahani Engineering College, Bandra, Mumbai, India
[2]Research Guide, Principal, Thadomal Shahani Engineering College, Bandra, Mumbai, India

***Abstract*—Organizations possess legacy software applications and systems, which are valuable assets and significant to current business operations. One of the main challenges in today's organizations is how to efficiently and effectively integrate legacy assets with other internal and external applications and business processes. If an organization has previously implemented enterprise resource planning (ERP) and other legacy systems and intends to integrate a business intelligence (BI) system, then BI and ERP integration might be challenging because of several factors. However, the challenges can be conquered with proper planning, and organizations can obtain massive benefits from their investments in ERP as well as BI systems. In this study, various integration benefits and integration approaches are elucidated. Moreover, this study explores and investigates contemporary software engineering challenges in integrating disparate software systems.**

***Keywords*—Challenges, integration, disparate software systems, enterprise resource planning, business intelligence**

I. INTRODUCTION

This study aims to investigate contemporary software engineering challenges in integrating disparate software systems, and the investigation method is based on objective data gathering and analysis. In addition, data gathering is based on literature reviews on enterprise integration. Successful software systems must be progressed or advanced to remain effective because they are increasingly modified in various ways and released afresh. The trend is to strengthen integration and interoperability possibilities of software systems with other systems, which is achieved by supporting open or effective standards through middleware.

The integration of legacy and other disparate systems from various vendors or developers has been viewed as a major issue in IT. For an organization, the integration of IT and systems is one of the most essential, complex, and expensive areas. Bernstein and Haas (2008) suggest that system integration is the "biggest and most expensive challenge" in IT, and evaluations by them indicate that integration costs 40% of IT large shop budgets [1]. System integration is a combination of disparate technology products that an organization uses for its operation, which requires interaction and communication among thousands of different hardware, software, communication, and process components. System integration has two faces. The first face is concerned about the internal activities of firms as they develop and integrate the inputs they require to produce new products. The second face—which has gained popularity recently—is concerned about the external activities of firms as they integrate components, skills, and knowledge from other organizations to specifically produce complex products and services. Integration generally provides a major challenge for today's organizations [2].

Enterprise resource planning (ERP) software records large data volumes across several business functions. The responsibility of business intelligence (BI) systems is to convert this data into relevant information that can be used for informed decision making. With BI software evolution and rapid changes occurring in the ERP software market, two singular systems are forming a natural symbiotic connection. ERP software pairing with BI solutions can provide end users their desired information in a comprehensible and actionable format.

To modernize business processes and increase productivity, organizations all over the world invest in systems software such as ERP, supply chain management, and other BI software. These disparate software systems are from different vendors, and they run on different platforms. The value of these software systems can only be understood if they are suitably integrated with one another. In today's competitive environment, organization's decision makers are concerned to stay well informed as to what is happening in their business and throughout the industry. Consequently, businesses to a greater extent are placing importance on BI and ERP system integration as a platform for informed decision making. Nevertheless, despite best intentions, many organizations struggle to effectively implement BI and ERP software systems. New application integration with existing enterprise information systems is challenging for many organizations worldwide. On the basis of systematic acquisition, collation, analysis, interpretation, and information exploitation, enterprises decision makers are in a better position to make suitable business decisions [3]. The integration of ERP and BI systems can enhance and improve the decision-making ability of organizations by leveraging the ability to manage data from the ERP system and the analytical competences of the BI system [4].

The remainder of this paper is organized as follows. Section 2 defines and discusses the concept of integration, Section 3 presents integration architecture, Section 4 describes the approaches of integration, Section 5 examines the challenges involved in integration, and Section 6 concludes the study.

II. INTEGRATION

Radatz et al. (1990) define integration as "the process of combining software and hardware components or both in a system" [5].

Several generations of systems that depend on various technologies developed over years are used by many organizations. These technologies (i.e., old as well as new) provide huge support to organizations. Unfortunately, most of these systems cannot communicate or share information with each other and other advanced systems. There is always an option of replacing these old systems with new ones, but it is very costly and time consuming. Moreover, these old systems contain lots of data and information, which are of high importance to organizations.

System integration can be viewed as a way to solve most of the abovementioned problems. System integration was initially confined to technical aspects such as for connecting computer hardware components. As knowledge on IT evolved, integration also appeared to be useful in software, data, and communication. Furthermore, integration is a complex task to facilitate unified information and data flow between different systems within and across enterprises.

An integration strategy can be classified into two types, i.e., intrusive and non-intrusive. An intrusive integration strategy—known as white box integration—is the one that requires the knowledge of internal interfaces of a legacy system, whereas a non-intrusive integration strategy—known as black box integration—is the one that requires the knowledge of external interfaces of a legacy system.

There are two main approaches for integrating legacy systems, i.e., application and data integration.

1) Application Integration

In this approach for integrating legacy systems, applications contain the enterprise's business logic, and the solution lies in preserving that logic by extending the application's interfaces to interoperate with other or new applications.

Application integration can be defined as an activity that integrates and standardizes an enterprise's isolated business applications, processes, and functions to provide common shareable business applications, functions, and services within the enterprise. It has the following advantages. First, it provides more functions and better services than individual systems. Second, it can reduce data redundancy and function overlapping, thereby ensuring a greater degree of data integrity and consistency [6].

2) Data Integration

In this approach for integrating legacy systems, the enterprise's real currency is its data. The implied business logic in the data and metadata can be effortlessly manipulated directly by applications in the enterprise's new architecture. Some data integration solutions are described below:

System integration architectures such as enterprise application integration (EAI) and Web services are popular in the market. Leading software companies such as SAP and IBS develop different types of software that support these architectures.

A. System Integration Eras

As per Johnson (2002), there are four system integration eras. First, there were the "stovepipe" systems, which were isolated and ill-equipped, and they hardly performed communication with their neighboring systems, but they contained valuable organizational data. These systems are also known as legacy systems. If any output was occasionally needed from one system to the other, then information transfer was performed manually [7].

As organizations computerized their business and more information was digitalized, the abovementioned system became unproductive. Systems that required communication with other systems were then preferably integrated. This era was known as the point-to-point era. As this approach was adopted, it became evident that system integration cost was high, and in many cases, using this approach was expensive. Furthermore, when the enterprise software system became larger, managing many customized connections between systems became difficult [7].

ERP systems were introduced to reduce the enterprise's software system complexity. After examining the likenesses of most computerized organizations, vendors such as SAP and Baan started offering huge systems, which cover most of the functions that previously needed to be performed separately. From an architectural perspective, one of the benefits of ERP systems was that components or systems were developed by one vendor and prepared for integration with each other. This was called the ERP era. Many ERP implementation projects failed because of several reasons such as poor adaptation to organizational needs [7].

Now the interest is for EAI solutions, which include message brokers, etc. These products are particularly designed for the facilitation of integrating the abovementioned legacy and ERP systems, which is known as the EAI era. Nowadays, many organizations have systems that belong to the abovementioned eras, and very few of them have managed to fit themselves into one specific era [7].

B. Significance of System Integration to Organizations

Organizations have data and applications that belong to different computational generations, which are written using different programming languages, and these languages have different vocabularies and syntax rules, data types, and other inconsistencies. Therefore, it is difficult to integrate these applications. Thus, crucial and valuable organizational data are held hostage [8].

C. Benefits of Integration

There are numerous organizational benefits, for example, increased profitability, decrease in costs, and increased efficiency, from system integration. Integration solutions enable using data and functionality exemplified in the existing applications of organizations or legacy systems instead of replacing these systems with new ones. In addition, these systems provide long-run benefits, for example, organizations can gain an instant and real-time view of all their data and operations, which help in making better decisions. Furthermore, they provide flexibility to rapidly adapt business processes to accommodate growth and meet new business challenges [9].

III. Integration Architecture

Herein, we discuss point-to-point and EAI system integration architectures.

A. *Point-to-Point Integration Architecture*

Connectors have to be built to obtain two independent systems to communicate with each other. These connectors must be able to translate data structures from one system to the other. In point-to-point integration, integration code for each interface is required for integrating systems. When any changes occur in any applications, the interface programs must be updated and changed. Furthermore, application integration becomes difficult as new applications are added. It is essential for a newly added system to create connection point interfaces with an existing system that it is connected with. Consequently, the integration solution grows in complexity and eventually becomes difficult to manage [8].

B. *EAI System Architecture*

Enterprises attempt to share data and processes without making comprehensive changes to the applications or data structures and correspondingly decrease the number of interface points, which is possible only through the EAI architecture.

"EAI is the creation of business solutions by combining applications using middleware" [10].

The EAI architecture consists of a central system, i.e., the hub, which acts as the middleware. In this approach, rather than the requestor communicating with the respondent, the requestor communicates with the hub, which in turn communicates with the respondent [8].

When application A needs data from application B, A sends a request to the hub using hub's language. The hub translates the request and sends this request to application B. Application B receives the request and converts the request in its own format. The adapter on application A understands only its and the hub's language. Likewise, the adapter on application B understands only its and the hub's language [8].

C. *Middleware*

Middleware is a type of software that facilitates communication between two or more software systems, for example, message broker. This is achieved by providing common interfaces, which in turn enables all integrated applications to pass messages to each other. Middlewares are mainly used for moving information between applications and databases [11].

IV. Approaches of Integration

There are several approaches of integration with ERP each having their own advantages and disadvantages.

A. *Point-to-Point Integration with ERP Interfaces*

Software applications can directly access the key interface. For convenience, many developers in search of prompt solutions to integrate directly connect their applications with ERP. The main advantage here is the initial time to write a single application against ERP. While point-to-point integration in some cases can have a low initial hurdle, it is normally the most expensive and burdensome method to integrate with ERP. In these types of architectures, applications become tightly coupled with ERP, which makes them weak as changes in the ERP environment changes with time. Furthermore, as business processes change and new integration scenarios are needed, integration becomes more and more complex as it involves supplementary touch points and tight dependencies. The end result is a weak environment that fails to deliver business value in the long run. Therefore, these types of integrations occur more than practical "hacks," and they are usually not suggested.

B. *Service-oriented Architecture (SOA) stacks*

The next approach of integration with ERP is SOA stacks. Large vendors such as IBM, Oracle, and others provide these integration solutions. SOA stacks consist of multiple products such as application servers, enterprise service buses, orchestration engines, management tools, and development tools. These stacks occasionally include half a dozen or more products. They are mainly used to create a comparatively robust integration architecture that can address most use cases. Here, applications are loosely coupled, and so when changes are required, these applications can be addressed rapidly. In addition, the maintenance costs of applications are lower as compared to the point-to-point approach, and the overall platform is more reliable. Furthermore, because SOA stacks can enable ERP when implemented once, the cost and complexity of applications to support new business processes is significantly less as compared to the point-to-point approach. SOA stacks have several disadvantages. They involve multiple products that must be deployed and configured. A complete SOA implementation using one of these stacks can take many years and unexpected additional cost. Meanwhile, new application development can grind to a halt. Finally, all developers must be trained on the vendors privately owned tools. In addition, recruiting new developers is difficult because specialized skills are needed, and these new developers must possess additional knowledge beyond ERP. Of course, many organizations struggle with implementing, maintaining, and recruiting a massive IBM or Oracle solution to integrate with their already sizeable ERP implementation.

C. *Lightweight Standalone Enterprise Service Bus (ESB)*

Standalone ESBs are the latest approach of integration with ERP. ESBs are one of the primary components of a SOA stack. However, unlike the legacy stacks, standalone ESBs can operate without any outside application servers or other infrastructure components. Furthermore, standalone ESBs provide their own management tools or integrate with any management tool that the organization uses. Finally, they use industry standard technologies and development tools that developers are already acquainted.

In most of the cases, a lightweight standalone ESB is the optimal solution for integration with ERP. These products permit organizations to service-enable ERP without a massive upfront investment, developer training program, or multi-year rollout. They allow organizations to meet their most business goals in a short period of time while simultaneously laying the foundation for the future growth. While contributing the advantages of a SOA stack, standalone ESBs reduce the upfront cost and risk associated with forming a solid foundation for various ERP integration scenarios.

There are various other approaches of integration with BI in order of increasing cost, delays, and risk.

- *Single source approach:* Buy BI from the same supplier that provides your ERP solution. There is also a possibility that complete integration is already built-in.
- *Pre-integrated approach:* Find a third-party BI supplier who has already developed an integration with your ERP brand and version.
- *Fourth-party integration approach:* Some independent or may be dependent software suppliers may have developed a packaged integration between your ERP and chosen BI. This may be a unique opportunity limited mainly for ERP and BI products with a large installed base.
- *Middleware approach:* Integration can be achieved via middleware toolsets using SOA and Web service design. This approach is helpful when there are numerous integrations to be built and maintained since the tools can be expensive.
- *Custom programming approach:* Best suited for one-to-one integration projects. They are highly expensive, difficult to maintain, and take a lot of time to build and test. They are less widespread than other approaches because the links are all handmade.

V. Challenges of Integration

Mature technologies exist that make integration manageable from an engineering perspective. However, these technologies have a lot of limitations, and hence, they provide opportunities for pioneering research. Although industry estimates differ, but approximately 70% of IT spending may be dedicated to integration-related activities. Consequently, it is not surprising that the technology world is overflowing with software solutions aimed at making integration a more manageable task [12].

An application has varied and many integration points and has many complex data relationships that need to be established and maintained. Thus, application integration is challenging.

The other challenge is difficulty in integrating multiple data models and instances of the application, which is a major concern for organizations that provide their application to different service providers or vendors. Data or process synchronization increases the level of complexity in these organizations [13].

Regrettably, enterprise integration is a very difficult task. It has to deal with multiple applications running on different platforms at different locations, making “simple integration” pretty much an oxymoron. EAI suites are offered by software vendors that provide cross-platform as well as cross-language integration and the ability to interface with many popular packaged business applications. However, this technical infrastructure presents only a small portion of the integration complexities. The true challenges of integration span across business and technical issues.

Though an organization might be successful in implementing ERP, various challenges may arise when attempting to integrate a BI solution. Table I lists the key challenges that are faced during integration.

TABLE I Challenges of Integration

Sr. no	Challenges	Descriptions
1	Data structure	It is easier to change data structure in BI than in ERP as it is indirectly tied to a business operation.
2	Upgrade implications	When planning for an upgrade of integrated ERP and BI systems, remember that the integration setup and infrastructure need to be updated. The current setup needs to be re-configured to meet post-upgrade requirements, and this could cause unexpected technical issues. Understanding upgrade implications and appropriate planning are the keys to ensure business continuity.
3	Selecting BI software: Use your ERP vendor or adopt third-party BI	Choose the existing ERP vendor’s BI system, which is often advertised as a more straightforward implementation. “Clients who are inclined to have similar environments with their ERP tend to adopt an ERP provider’s solution; however, that is only possible if they are similar in terms of competence. The advantages of combining BI and ERP vendors would be easy integration, packaged licensing costs, lower system ownership costs, familiarity with vendor interfaces, etc.
4	Integration with third-party BI	A more proficient third-party solution can obtain the permission if the ERP vendors’ solution is constraining. “If you require advanced analytical capabilities that BI offered by ERP vendors do not provide, then it is rational to use a third-party BI.
5	Integration technology	Different integration techniques require various specialized software and hardware, which can be expensive, can cause proprietary lock-in, and increase developer burden in understanding how tools can be used to integrate applications.
6	Limited control	In most of the cases, applications are “legacy” systems or packaged applications that cannot be changed to be simply connected to an integration solution.
7	Interoperabil-ity	Interoperability is the capability of two or more software components to communicate and cooperate with each other regardless of differences in language, interface, and execution platform [14, 15]. For this, components need to have similar understanding of their interface.
8	Diverse architectures	ERP system design and other primary applications are different. In addition, even when the data is in a compatible format, it is difficult to obtain products to effectively integrate with each other.

Sr. no	Challenges	Descriptions
9	Evaluation	Different enterprise integration solutions provide several functions and features, which results in a complex evaluation process. To overcome this, appropriate tools for evaluating the core integration features of enterprise integration solutions are required.
10	Data or functionality	Integrated applications may not wish to share data, but they may wish to share functionality such that each application can invoke the functionality in other applications. Invoking functionality can be difficult, and even though it may appear to be the same as invoking local functionality, it works differently with significant consequences for how well the integration works.
11	Security	There exist some security issues that need to be addressed. We must focus on the security mechanisms of legacy systems.
12	Constraints	Some of the constraints present in integrating legacy systems are on how to deal with components, connectors, semantics, and topology.
13	Data format	Integrated applications must select the data format they exchange or must have an intermediate translator to unify applications that insist on different data formats. A related issue is data format evolution and extensibility, i.e., how the format can change with time and how this change will affect the applications.
14	Integration style	*File transfer:* Does each application produce files of shared data for others to consume and consume files that others have produced? *Shared database:* Do applications store the data they wish to share in a common database? *Remote procedure invocation:* Does each application expose some of its procedures so that they can be invoked remotely and do applications invoke those to run behavior and exchange data? *Messaging:* Does each application connect to a common messaging system and exchange data and invoke behavior using messages? The trick is not to select the one style to use always but to select the best style for a particular integration opportunity. Each style has its advantages and disadvantages. Two applications may integrate using many styles such that each integration point benefits of the style that suits it best. Similarly, an application may use different styles to integrate with different applications in order to choose the style that works best for the other applications. Some integration approaches can best be viewed as a hybrid of many styles. An integration product or EAI middleware may employ a combination of styles, all of which are effectively hidden in the product’s implementation.
15	Right integration architecture	Architectural issues comprise gross organization and global control structure; protocols for communication, synchronization, and data access; assignment of functionality to design elements; physical distribution; design element composition; scaling and performance; and selection among alternatives (an architectural style).
16	Testing	There are several vital differences. First, integration project architectures are message based and not code based, and the failure points are usually in messages and not in codes as these points would be in an application development project. Unfortunately, most testing tools focus on testing codes and graphical user interfaces and not on testing messages.
17	Selection of an appropriate integration tool	The following factors need to be considered when selecting an appropriate integration tool: 1) cost 2) speed 3) user interface 4) scheduling 5) scalability customization options
18	Integration approach	There are several integration approaches. More evaluation is needed to authorize or support the appropriate approach.
19	Next-generation capabilities	Even with improvements on the technical aspect, ERP system integration tackles new obstacles such as cloud, mobility, and new data types. ERP integration remains an immortal challenge for IT organizations as they enhance enterprise systems to support a plethora of next-generation capabilities.
20	Use of right tools	It is essential that the tools chosen can improve and simplify the integration process. Moreover, these tools should be user friendly, test, configure, debug, and deploy.
21	Beginning of open source BI	Beginning of open source BI (OSBI) has added a new dimension to the BI landscape. OSBI allows faster returns on smaller investments. Mainly driven by cost considerations, interest in and adoption of OSBI within organizations as a whole at those using ERP is pervasive and growing.
22	User manuals and system design documentati-on are outdated	Legacy system architectures are poorly documented, which leads to a new type of problem when integrating a legacy system with an overall system architecture design and specification.
23	Integration mechanism	What type of integration mechanisms exists in the application? Many in-house legacy systems are most likely to be built assuming little or no outside connectivity with other applications. What type of integration mechanisms exists in the application being targeted? If integrating with a commercial application, development environment, or modernization tool, integration capabilities are usually built-in.
24	Real time	What are real-time data requirements? Nowadays, although there are many system integrations that need to be real time, some systems need not be real time. Invoicing every night means integrations can be performed in a different way if they are sent on demand. A business application that requires online ordering over the Web has different real-time data requirements than an internal application with a smaller user set without time-critical data.
25	Data integrity	Adeptness to certify that data is consistent between applications anytime with entailed performance metrics.

VI. Conclusion

System integration is a convoluted area where several factors need to be taken into account. Moreover, executing an efficacious integration requires knowledge and understanding of the requirements weighed against a careful balance of conciliations. In this study, we discussed twenty-five challenges of integration for disparate software systems.

References

[1] P. A. Bernstein and M. H. Laura, "Information integration in the enterprise," *Communications of the ACM*, vol. 51, no. 9, pp. 72–79, 2008.

[2] A. Peslak, "A study of information technology integration," *Journal of Information Systems Applied Research*, vol. 4, no. 2, p. 19, 2011.

[3] D. Singh and P. Kumar, "Design of integrated business intelligence system framework for insurance business processes," *International Journal of Applied Information Systems*, vol. 3, no. 3, pp. 42–48, 2012.

[4] M. I. Nofal and Z. M. Yusof, "Integration of business intelligence and enterprise resource planning within organizations," *Procedia Technology*, vol. 11, pp. 658–665, 2013.

[5] J. Radatz, A. Geraci, and F. Katki, "IEEE standard glossary of software engineering terminology," *IEEE Std 610121990*, no. 121990, p. 3, 1990.

[6] H. Moradi and A. Bahreininejad, "Toward a comprehensive framework for evaluating the core integration features of enterprise integration middleware technologies," *Journal of Systems Integration*, vol. 4, no. 1, pp. 13–29, 2013.

[7] P. Johnson, "Enterprise software system integration: An architectural perspective," 2002.

[8] B. E. Travis and M. Ozkan, *Web Services Implementation Guide*, Architag Press, Denver, CO, 2002.

[9] InterSystems Corporation (2004). Integration Technology Primer. [Online]. Available: http://www.intersystems.com/our-products/ensemble/ensemble-technology-overview/

[10] W. A. Ruh, F. X. Maginnis, and W. J. Brown, *Enterprise Application Integration: A Wiley Tech Brief*, T. Hudson, Ed. John Wiley & Sons, 2002.

[11] D. S. Linthicum, *Enterprise Application Integration*, Addison-Wesley Professional, 2000.

[12] I. Gorton, D. Thurman, and J. Thomson, "Next generation application integration: Challenges and new approaches," *In Computer Software and Applications Conference*, 2003. COMPSAC 2003. Proceedings. 27th Annual International, pp. 576–581, IEEE, 2003.

[13] F. Hojaji, "Five aspects of application integration requirements," *ARPN Journal of Systems and Software*, vol. 2, no. 3, pp. 97–103, 2012.

[14] P. Wegner, "Interoperability," *In ACM Computing Surveys*, vol. 28, no. 1, pp. 285–287, 1996.

[15] J. C. Wileden and A. Kaplan, "Software interoperability: Principles and practice," *In Proceedings of 21st International Conference on Software Engineering*, pp. 675–676, ACM, 1999.