Challenges and Future of Enterprise Application Integration

Tariq Rahim Soomro
College of Engineering & IT
Al Ain University of Science & Technology, Al Ain, UAE

Abrar Hasnain Awan
Department of Information Technology, SZABIST Dubai Campus, UAE

ABSTRACT

In today's rapid growing business world, organizations are relying on technology and need of the integration of the disparate applications is in high demand. Enterprise Application Integration (EAI) is use of hardware and software to integrate a set of Enterprise Computer Applications. This paper will review the types of EAI along-with its architecture. Challenges in the field of EAI will be discussed later and finally future of EAI will be discussed.

Keywords

EAI, Challenges of EAI, Future of EAI and ESB.

1. INTRODUCTION

Today the organizations heavily rely on the technologies which are of different nature and the applications running in different business units based on different technologies. So the integration of these applications into a unified set of business process has emerged as a priority. Enterprise Application Integration (EAI) provides a mechanism to share both processes and data. In the competitive business environment to shorten the application development life cycle and to reduce the financial cost the reuse of existing database and application is a first choice instead of recreation. The organization systems are based on new and old technologies and use the broad range of platforms, databases and application, and based on different programming languages. Forester Research estimates that up to 35 percent of development time and 30 percent of development cost is spent to creating interfaces and points of integration for applications and data sources. Therefore, the solution is the EAI; as it is the collection of technologies and approaches to provide short-term and cost-effective solution. It is a technology-business philosophy where all enterprise systems can share information and logic in easiest way. Using EAI applications share data and process without changing the applications and their data structures which is cost effective [1]. According to [2] EAI is "the combination of processes, software, standards, and hardware resulting in the seamless integration of two or more enterprise systems allowing them to operate as one". EAI is the solution for the applications developed without central vision or strategy and served a single purpose. The applications can be for example, inventory control, sales automation, general ledger, and human resource. The applications are built for the specific need and based on the technology-of-the-day, built on non-standard approach and can-not interoperate. The old applications can be replaced with the application based on latest technologies but adds additional financial overhead to the organization [1]. Following are few advantages of EAI [3][4]:

- It helps to access the real time information among the applications
- It simplifies the business process to improve the efficiency of the enterprise
- It provides the integrity of the information
- It provides the easy development and maintenance
- It improves the customer relationships
- It improves the supply chain relationship
- It keeps legacy applications alive
- It improves the business process and time to market is reduced
- It supports the business and organizational changes in some extent
- It brings the standard approach in applications
- It brings responsive technology to changing business needs
- It reshapes the applications according to current and future business needs

This paper organized as follows, section 2 will review types of EAI; section 3 will review its architecture; section 4 will review the challenges and problems of EAI and final section will discuss EAI along-with future work in the field of EAI.

2. TYPES OF EAI

The Enterprise Application Integration can be performed on different levels as following and shown in Figure-1 bellow [1]:

- Data level
- Application Interface level
- Method level
- User Interface level

2.1 Data-level EAI

Data-level EAI is related to the data integration, this technique is used in typical EAI-enabled enterprise. The data is moved from one data store to another and processes the information if required. The main advantage of this approach is that there is no code change which saves the cost of redevelopment [1].

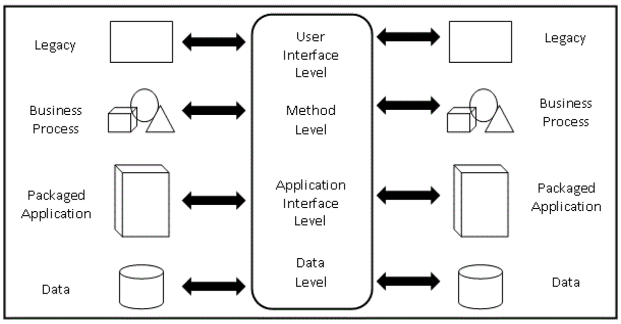

Fig 1: Types of EAI [1]

2.2 Application Interface-level EAI

The business process and data is accessed through the interfaces with this approach. The applications can be bundled together and share the business logic and data but the application interfaces have specific features and functions. This approach is mostly used with the packaged applications. Message brokers are used here as preferred alternative solution [1].

2.3 Method-level EAI

Method-level EAI shares the business logic. A method can be access by many applications and applications can access each other's method. It can be achieved in many ways, including distributed objects, application servers, TP (transaction processing) monitors, frameworks, and reusing the existing applications for creating new application. This approach is being used for long period as we reuse the business logic to reduce the development efforts. But this approach is not very effective due to technological and human constraints, and also due to the different needs of the problem domain [1].

2.4 User Interface-level EAI

This approach bundles the applications use their user interfaces. The mainframe applications can be accessed through this approach. This is not a preferred approach but is being used for years [1].

3. EAI ARCHITECTURES

EAI can also be categorized according to its design structure. Following are known architectures of EAI [5]:

- Point-to-Point topology
- Hub-Spoke topology
- Bus Topology

3.1 Point-to-Point Topology

Point-to-Point (P2P) considered as a traditional integration approach. It allows one application to connect to one application using a pipe. An application communicates with its pair using a procedure call or message. A connector is implemented for each pair of applications to communicate. The connector is responsible for the data transformation and integration for specific pair. It is possible to connect more than two applications but it involves too many complexities. This model is suitable for small organizations where few applications are integrated; it can be easily build and deploy for the customized needs for small infrastructure and if the number of applications increased, the connections are exponentially increased [1][6][7].

3.2 Hub-Spoke Topology

Hub-Spoke integration topology has centralized broker (Hub) and adapters (Spoke). Spoke is a connector which connects the application to the Hub. It translates the data for application and Hub to communicate. It transforms, translates and routes the messages for the destination. Adapter publishes messages to the message broker. The centralized Hub can be managed easily but it is not scalable large number of messages. Federated hub and spoke architecture is designed to overcome the scalability issue. It allows the loose coupling between the applications, so the applications can communicate asynchronously, because of its central style the integration configuration is less redundant [6][8].

3.3 Bus Topology

Most of the project adopted the Hub-Spoke model were not successful because of the lack of standards, expensive proprietary solutions, being heavyweight and worked for the homogenous. The centralize nature of broker model is the single point of failure, so the problem of single component causes the failure of entire network. The Bus model emerged as a solution to overcome the problems of broker model. It also used the centralized routing component but it distributes the tasks to others. These components can be grouped and hosted on different places in the network, and can be duplicated geographically for scalability. Other functionalities like security transaction processing, error handling, has been identified as the bus model evolved. The Bus model provided these functions can be enclosed in separate components. The Bus model becomes a lightweight, suitable and reliable integration solution. It is abstracted solution, having consistent pattern, and can be designed with less coding and no modification to the applications. The bus model finally came to known as Enterprise Service Bus (ESB) [6][8].

4. CHALLENGES AND PROBLEMS OF EAI

EAI provides common procedure of communication between applications and provides integration for business process and data. EAI provides the integration of business process and data, and it also provides the mechanism of reusability and distribution. On the one hand it does not demand high technical skills for the integration and on the other hand traditional middleware only provides the integration of data and requires special set of technical skills [1]. Enterprise application integration (EAI) is a complex undertaking, and the following six pitfalls can bedevil the process [7]:

- A shortage of limited skills;
- Lack of recognition that EAI is an architecture, not a product;
- Neglecting security, performance and monitoring;
- Implementing EAI as part of another project;
- Going ahead without an integration strategy;
- Internal politics and poor communication.

It is reported that 70% of EAI project failed due to the management issues. According the Chairman of Integration

Consortium European Mr. Steve Craggs these seven main problems currently faced by the companies as they implemented EAI solution and these are [3]:

1. Constant change
2. Shortage of EAI experts
3. Competing standards
4. EAI is tool paradigm
5. Building interfaces is an art
6. Loss of detail
7. Accountability

Other researchers identified some more problems as follows [3]:

1. Emerging requirements
2. Protectionism

According to [4][9] the cost of development of EAI is higher than traditional approach. Implementation takes more time and consumes more resources. EAI is complex to build because of existing enterprise architecture and the EAI has shortage of required skills. EAI provides more access so, the security is a big concern as the business and IT changes rapidly.

5. DISCUSSION AND FUTURE WORK

It was reviewed that how EAI is evolved as information-oriented (data level), to interface-oriented (application interface), to process-oriented (method level) and to service-oriented as “Enterprise Service Bus”, which is the latest approach developed so far [10]. Currently Enterprise Service Bus (ESB) is the most useful architecture to design and implement EAI solution [11][12][13] but the EAI is still a challenging area [14] and there is no ideal approach and/or technology exist to build an integration solution [3]. As discussed in challenges and problem section that EAI still lack in standardization [15], but no doubt the Enterprise Integration Patterns (EIP) [16] is accepted and adopted as de-facto messaging standards in the integration solution by researchers [17]. As an future initiative OpenEAI project started as an initiative to defining messaging structures and integration standards [10][18]. Keeping in view the context of challenges, for example, constant change, shortage of EAI experts, building interface is a complex and difficult and monitoring. Integration Platform as a Service (iPaaS) is a cloud based integration solution [19][20] to design, build, monitor, and manage integrations centrally and especially addresses the maintenance challenge of distributed integration solution and benefits the future adoption of SaaS model [21]. Another example is iON – a cloud based iPaaS provides cloud-to-cloud and cloud-to-premise integration [20]. Based on the problems faced by EAI, for example, vendor lock-in of proprietary based solution, there are open projects by independent researchers as well as academic projects. The proprietary based solutions create a vendor lock-in due to non-standard implementation [22]. The shortage of EAI experts, skill set, development complexity of integration solution, rapid changes in the business etc. demands a platform independent abstract solution. The academic project, for example, Guarana DSL suggests EAI solution as, a platform and technology independent high level abstract design with MDA (Model Driven Architecture) approach which can automatically transform Platform Independent Model (PIM) into a Platform Specific Model (PSM) [3]. No doubt the future of EAI is based on platform independent based solutions, but lack of standards, shortage of EAI experts, constant changes, building complex interfaces and security in terms of performance and monitoring are to be reviewed.

6. REFERENCES

[1] David S. Linthicum, 1999, Enterprise Application Integration, Addison Wesley Professional, 1st Edition, ISBN-10:0201615835, ISBN-13: 978-0201615835

[2] James Fenner, Enterprise Application Integration Techniques, http://www.cs.ucl.ac.uk/staff/ucacwxe/lectures/3C05-02-03/aswe21-essay.pdf, Retrieved September 2011

[3] TIBCO Guru, February 2011, Enterprise Application Integration (EAI), http://www.tibcoguru.com/enterprise-application-integration-eai/

[4] William Tse, Enterprise Application Integration (EAI), http://www.cs.ucl.ac.uk/staff/ucacwxe/lectures/3C05-03-04/EAI.pdf, Retrieved November 2011

[5] Bc. David Kusak, 2010, Comparison of Enterprise Application Integration Platforms, http://zamecnik.org/_media/xrouter/eai_platforms.pdf

[6] MuleSoft Community, Understanding Enterprise Application Integration - The Benefits of ESB for EAI, http://www.mulesoft.org/enterprise-application-integration-eai-and-esb, Retrieved 2011

[7] Alex Nghiem, August 2002, Web Service Part 6: Models of Integration - Point-to-Point Integration, http://www.informit.com/articles/article.aspx?p=28713&seqNum=2

[8] Anurag Goel, Enterprise Integration EAI vs. SOA vs. ESB, http://hosteddocs.ittoolbox.com/Enterprise%20Integration%20-%20SOA%20vs%20EAI%20vs%20ESB.pdf, Retrieved November 2011

[9] IBM-EAI, April 2009, Enterprise Application Integration, http://ibm-eai.blogspot.com/2009/04/enterprise-application-integration.html

[10] David S. Linthicum, October 2001, Merctor Next Generation Application Integration, From Information, to Process, to Services, http://bukovec.fei.tuke.sk/predmety/pdt/referaty/EnterpriseApplicationIntegration/Steller/zad1/wp_next_gen_app_integration.pdf

[11] J. Davies, D. Schorow, and D. Rieber, 2008, The Definitive Guide to SOA: Enterprise Service Bus, 2nd Edition, APRESS, ISBN: 1430210575, ISBN-13: 9781430210573

[12] P. Delia, and A. Borg, 2008, Mule 2: Official Developer's Guide to ESB and Integration Platform, APRESS, Springer-Verlag New York Inc., ISBN13: 9781430209812, ISBN10: 143020981X

[13] D. Woolston, 2007, Foundations of BizTalk Server 2006 APRESS, ISBN: 1590597753

[14] Rafael Z. Frantz, Rafael Corchuelo, and Jesu Gonzalez, 2008, Advances in a DSL for Application Integration, In Web Application Integration (ZOCO) in JISBD, Vol. 2,

No. 2, 2008, www.sistedes.es/TJISBD/Vol-2/No-2/articles/Frantz.pdf

[15] Joe McKendrick, September 2009, ESBs are Patterns, not products, http://www.zdnet.com/blog/service-oriented/esbs-are-patterns-not-products/2878?tag=search-results-rivers;item3

[16] Gregor Hohpe, and Bobby Woolf, 2003, Enterprise Integration Patterns, 1st Edition, Addison Wesley, ISBN-10: 0321200683, ISBN-13: 978-0321200686

[17] Bitpile.com, 2011, Application Integration, http://www.bitpipe.com/tlist/Application-Integration.html

[18] OpenEAI, 2011, About the Project: OpenEAI Overview, http://www.openeai.org/live/index.php?p=5&sub=1

[19] Interarbor Solution LLC, 2011, MuleSoft takes full-service integration to the cloud with iON iPaaS ESB Platform, http://briefingsdirectblog.blogspot.com/2011/06/mulesoft-takes-full-service-integration.html

[20] MuleSoft, 2011, iON – Integration Platform as a Service, the simplest way to integrate SaaS and cloud, http://www.mulesoft.com/mule-ion-ipaas-cloud-based-integration-demand

[21] ZDNet, 2010, Taking enterprise application integration in the cloud, http://www.zdnet.com/news/taking-enterprise-application-integration-into-the-cloud/422936

[22] Fuse ESB, 2011, Chapter 1. Introducing Fuse ESB, http://fusesource.com/docs/esb/4.4.1/esb_prod_intro/ESBGetStartedOverview.html