A model driven architecture for enterprise application integration

Link to publication record in Manchester Research Explorer

A Model Driven Architecture for Enterprise Application Integration

Adra Al Mosawi[1], Liping Zhao[2] and Linda Macaulay[3]
School of Informatics, The University of Manchester
[1] Adra.Al-mosawi@postgrad.manchester.ac.uk
[2] liping.zhao@manchester.ac.uk
[3] linda.macaulay@manchester.ac.uk

Abstract

To achieve greater automation of their business processes, organizations face the challenge of integrating disparate systems. In attempting to overcome this problem, organizations are turning to new integration software called Enterprise Application Integration (EAI). Implementing EAI is a complex task involving both technological and business challenges and requires appropriate EAI architecture. This paper first provides an overview of current approaches to enterprise application integration together with an assessment of their strengths and weaknesses. It then proposes a new enterprise application architecture, based on the idea of OMG's Model Driven Architecture, to structure the EAI problem into five general types of model. This architecture is developed in response to the need to separate the technological aspect from the business aspect so that the both can evolve independently. The success of EAI lies in its resilience to the both technological and business changes.

1. Introduction

Enterprise Application Integration (EAI) can be defined as an activity that integrates and harmonizes an enterprise's isolated business applications, processes and functions in order to provide common, sharable business applications, functions and services within the enterprise. There are many obvious advantages with EAI. First, it offers more functions and better services than the individual systems. Second, it can reduce data redundancy and function overlapping, hence ensuring a greater degree of data integrity and consistency [18].

Enterprise application integration is a complex task involving both technological and business

challenges. From the technological viewpoint, EAI involves integration of incompatible and heterogeneous technologies. From the business viewpoint, EAI involves integration of disparate business processes and functions. The rapid change of business and the need for an enterprise to communicate and share information with other enterprises means that EAI is an on-going challenge that every enterprise will face.

Over the past decade, specialised software tools, termed as EAI solutions, have been developed to support the integration of an enterprise's application systems. These EAI solutions provide the infrastructure and mechanisms for extending and integrating both old and new application systems. They also provide the facility for the Internet access of an enterprise's applications and for enterprises to share the information and applications. Yet, EAI solutions vary in their underlying approaches and adopt different terminology and concepts. The diversity of the enterprise application integration approaches has made selection of an EAI solution difficult. Consequently, the more the enterprise application integration approaches, the more the EAI solutions, the harder it is for an enterprise to find the most suitable solution.

After a detailed study of the existing enterprise application integration approaches, we have observed that they can be classified into two general categories. The first category consists of the enterprise application integration approaches that reduce the integration problem to one particular part of the system such as data or process. For example, Linthicum [8] proposed four integration approaches, which are data integration, application interface integration, method integration, and user interface integration; Pushmann and Alt [12] suggested three approaches corresponding to data, object and process integration.

The second category consists of the enterprise application integration approaches that take an architectural view and address the integration problem from many aspects. For example, Duke et al. [3], partitioned the integration problem into different architectural layers, including the business layer, application layer and technological layer. Brown [1] split Duke et al's [3] business architectural layer into business process and information architectural layers.

This paper thus has a twofold purpose. First, it provides a comparison of the existing enterprise application integration approaches using the aforementioned categories and discusses their similarities, differences, strengths, and weaknesses. Second, the paper proposes a new EAI architecture based on the idea of OMG's Model Driven Architecture [11]. The proposed architecture captures the areas of EAI into five general types of model. The five model types are technology specific model, transaction service model, generic application service model, intra-application model, and inter-application model. This architecture is developed in response to the need to separate the technological aspect from the business aspect so that the both can evolve independently. Such a need arises from the very notion of EAI: the success of EAI lies in its resilience to the both technological and business changes.

The remainder of this paper is presented in four sections. Section 2 provides an overview and comparison of the existing enterprise application integration approaches. Section 3 describes our new EAI architecture. Section 4 discusses the data derived from the case study. The paper concludes in Section 5.

2. Overview of Enterprise Application Integration Approaches

2.1 Category One

Category one consists of the enterprise application integration approaches that focus on one particular area, such as data or processes. Often, some of these approaches can be combined in an EAI solution. For example, integration of object-oriented database systems will involve the integration of data, objects and processes. We have identified the following nine enterprise application integration approaches in this category, summarised in Table 1.

Data integration: This is perhaps the most common integration approach. It migrate data

between multiple data sources and allows data to be exchanged and shared between multiple organizations, applications and resources ([8], [14], [15]).

Object integration: This approach applies to the systems that are developed using object-oriented technology. It allows business object to cooperate and share functionality by creating reusable interfaces across applications [13].

Function or method integration: This approach solves the integration problem at the method or function level. It based on sharing business logic that exists within the enterprise ([8], [14], [15]).

User interface integration: This approach is the most primitive form of integration. It based on developing a user interface that imitates end user actions by using screen scraping or advanced terminal emulation ([8], [15]).

Table 1. Enterprise application integration approaches in category one

Integration Approach	Author
Data	Linthicum [8]; Pushmann and Alt [12]; Ring and Ward-Dutton [13]; Samtani et al. [15]; Ruh et al. [14]
Object	Pushmann and Alt [12]; Ring and Ward-Dutton [13]
Function/Method	Linthicum [8]; Ruh et al. [14]; Samtani et al. [15]
User interface	Linthicum [8]; Samtani et al. [15]
Application interface	Linthicum [8]
Presentation	Ruh et al. [14]
Process	Pushmann and Alt [12]; Samtani et al. [15]
Internal process	Ring and Ward-Dutton [13]
Cross-enterprise process	Ring and Ward-Dutton [13]

Application interface integration: This approach solves application interoperability through sharing of common business logic, exposed using a, pre-defined, programming interface. It based on exposing an interface from packaged, or custom, applications to invoke their business services or retrieve their data[8].

Presentation integration: This approach is the simplest form of integration. It allows creating new

user interface by re-mapping the old presentations [14]. This approach corresponds to user interface integration approach.

Process integration: This approach treats processes as common abstractions of the systems. It works at a higher level than objects and methods. This approach is also called business method by Samtani et al. [15]. It involves the flow of information and automation of business processes based on business rules of divers business systems [15]. This approach enables the coordination of decisions and the management of the dependencies between enterprise processes.

Internal process integration: This approach applies to manage organization processes internally. It allows to track the status of business process activities internally [13].

Cross-enterprise process integration: This approach is more suitable for B2B application integration. It allows to share processes among multiple B2B corporate entities [13].

To demonstrate fully the benefits offered by the aforementioned enterprise application integration approaches, we have summarised the strengths, and the weaknesses of the various enterprise application integration approaches in Table 2.

Data integration is a simple and inexpensive approach to implement [8]. It supports a unified view of data for all the organization units and the trading partners. Data can be moved between different business domains without requiring changes to the source, or target, database, and data can be shared, regardless of the database systems, operating systems or networking platform. This approach provides access to a wide range of data and allows data to be reused across other application. However, this approach assumes bypassing of the application logic, therefore limiting 'real time' transactional capabilities [15].

Object integration encapsulates the business logic and data within objects allowing the objects to be linked together in a plug and play manner to interoperate. This approach holds the greatest value for the enterprise of heterogeneous systems and architecture. It allows the legacy system to be wrapped into objects, and participate in the object integration[15]. However, this approach is complex and the most difficult approach [8].

Process integration is more advanced than data and object integration, as the logic, and reasoning, for conducting business are included. It removes the data integration limitation by focusing on the actual process and not only on the data [15]. Although process level integration offers the greatest business benefits, it is a complex, and expensive approach to implement. It requires a high level of investment and is complex to architect and develop [15].

Table 2. Overview of the strengths and weaknesses of EAI approaches

Integration Approach	Strength	Weakness
Data Shares data between multiple data sources	Simple to implement Inexpensive to implement Consistent data Minimum changes to the source or target applications Provides access to wide range of data	Does not invoke business logic
Object Integrates objects distributed throughout the enterprise	Integrate business logic Reusable	Complex Difficult approach
Process Business process modeling and integration	Real time tracking and analysis of business process Dynamic process optimization and adjustment Process evaluation	Complex to architect Expensive to implement
User interface Build new interface by re-mapping old interface	Easy to implement Requires minimum change to existing application	Tightly coupled application Synchronous communication Difficult to maintain Cannot scale Doesn't take place at actual interconnection
Application interface Exposes an interface for sharing business functionality	Invokes application functionality	Tightly coupled application Synchronous communication Require same API for both application

User interface integration approach is the easiest, and the faster, way of integration. However it has many limitations. This approach is synchronous in its communication style, and requires the original application to be online [8]. It cannot be scaled, and thus cannot handle more than a few screen interfaces at a given time [8]. It only takes place at the

presentation, and not at the actual inter-connection between applications and data [14].

Application interface integration allows external applications to access existing application services without modifying the existing application. However, this approach has many limitations; it is synchronous in nature where the function call is not returned until the specific action is completed, and requires both applications to be available, which may result in a tightly coupled system [9]. There are no industry standards for creating an API, and to enable API approach integration, both applications must use the same language or protocol [15].

2.2. Category Two

Enterprise application integration approaches in this category structure the integration problem into different architectural layers. We have identified the following 8 layers in this category, summarised in Table 3.

Table 3. Enterprise application integration approaches in category two

Integration Layer	Author
Business	Duke et al.[3]; Hasselbring [6]
Business process Information	Brown [1]
Inter-organizational	Hasselbring [6]
Application	Duke et al. [3]; Brown [1]; Hasselbring [6]
Enterprise application	Hasselbring [6]
Technology	Duke et al. [3]; Brown [1]; Hasselbring [6]
Middleware integration	Hasselbring [6]

Business architecture layer defines the enterprise structure and the workflow for business concepts, processes, and rules. Brown [1] split this layer into business process layer and information architecture layer, which are discussed below.

Business process layer defines the core processes that a business carries out to conduct a business.

Information architecture layer defines the subject areas that are of interest to business and the types of data required for assembling information in each subject area.

Hasselbring [6] expand this architecture layer by adding inter-organization layer to horizontally interrelate between business processes of co-

operating units. *Inter-organizational processes layer* seeks to organize business process in a competitive way by horizontally cutting business process through traditional organization structure.

Application architecture layer defines the implementation strategies for the business architecture layer. Hasselbring [6] expand this architecture by adding enterprise application integration layer for horizontal integration. *Enterprise Application Integration layer* seeks to integrate independent enterprise resource planning systems.

Technology architecture defines the infrastructure for business information and communication flows.

Hasselbring [6] expand this architecture by adding middleware layer for horizontal integration, to address the technique for integrating componentized information systems with state of the art technologies such as CORBA and database gateway.

3. A Model Driven EAI Architecture

In addressing the dimensions of application integration, this paper proposes a novel architecture based on the idea of OMG's Model Driven Architecture [11]. The proposed architecture is developed in response to the need to separate the technological aspect from the business aspect so that the both can evolve independently. Insulating business functionality and behaviour, from the technology specific code, permits the flexibility, and interoperability, needed for application integration. Thus, the proposed architecture captures the areas of EAI into five general types of models. The five models types are technology specific models, transaction service model, generic application service models, intra-application models, and inter-application models as in Figure 1. This architecture allows business domains to inter-act, and co-operate, through a set of common services. Both business domains and common services rely on technological facilities, which enable physical connection and the inter-action of the model.

Inter-application Model: defines the information flow and the relationships that exist between the organization and other trading partners such as customer, supplier and retailers. It determines the needs of partners companies with which it will be cooperated. It focuses on how business partners refine their business process so the applications supporting their business can be cooperated.
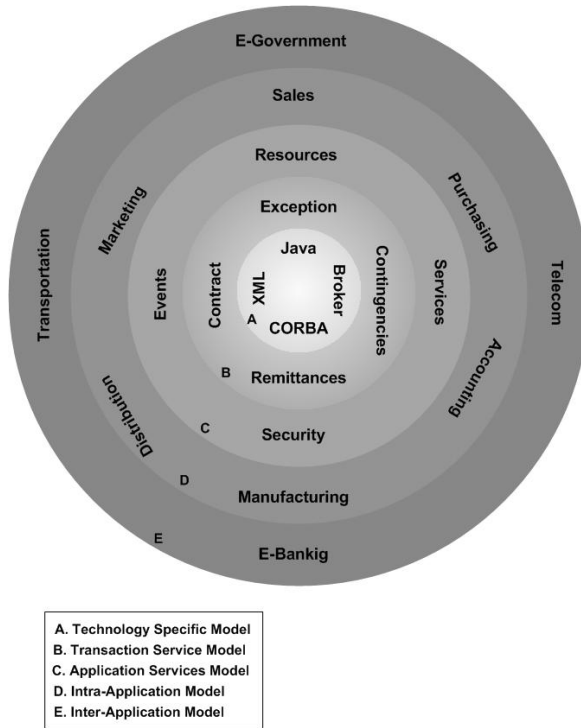
Figure 1. A Model Driven EAI Architecture

Intra-application Model: defines the functionality and the workflow to be delivered by the application service model. It defines the enterprise structure and the processes the enterprise carries out in conducting business.

Generic Application Service Model: defines the services required to deliver the required automation of business processes. These services support the management of functionalities and the operation of business processes, while hiding required detail necessary for a particular platform. This model comprises two types of services, which are discussed below.

Specialized services provide business service, and business logic, for a particular vertical domain, such as industry, financial or government applications.

Generic services represent generic services common to multiple vertical domains, such as order fulfilment, shipping or purchasing.

Thus, this model supports the function of both the individual business domain, and the definition, and management, of information and activities relevant to multiple business domains. The business services inter-act and inter-relate in a predictable manner to produce a recognizable business activity of a generic nature in a specific business domain.

Transaction Service Model: defines the transactional support for application services such as transfer of funds, invoicing, billing and payment. It defines contractual obligations, exceptions, contingencies, remittances and contract violations.

Technology Specific Model: defines the basic technological platform for the physical connection and inter-action of the application system. It specifies the communication infrastructure, communication technologies, network topology, modes of connectivity and distribution requirements.

The proposed architecture address the need to approach the integration from a business centric view because if the business integration needs driven by integration products capabilities rather than business need result in a tight coupling between business requirement for integration needs and the selected integration product. Therefore, there is a need to interspersing business requirements from implementation technology. Although this integration architecture is essential, and valuable, it is important to know that the business processes of co-operating units may not be inter-related. When requirements call for a process that slices horizontally across business functions, a horizontal integration of the models is required to support integration of business processes across the different business units. To inter-relate and handle business processes of co-operating units, there is, therefore, a need to consider horizontal integration. Two types of integration can be achieved by integrating the models horizontally: inter-application integration and intra-application integration as in Figure 2.

3.1. Inter-Application Integration

Inter-application integration allows enterprise to integrate its business process with its business partner's process, to improve business efficiency. To achieve inter-application integration, business processes between partners need to be co-ordinated, and defined, by developing an infrastructure that encompasses relationships between decision-makers, employees, customers, suppliers and partners. This infrastructure allows inter-operability between different business entities and allows for tracking transactions across different enterprise boundaries. This integration also enables businesses to communicate efficiently over the Internet, hence reducing transaction costs, shorten cycle times, enhance customer services and reduce marketing time. RosettaNet, BizTalk and EbXML are some standard technologies that provide common mechanisms, and semantics, for inter-application integration [9].
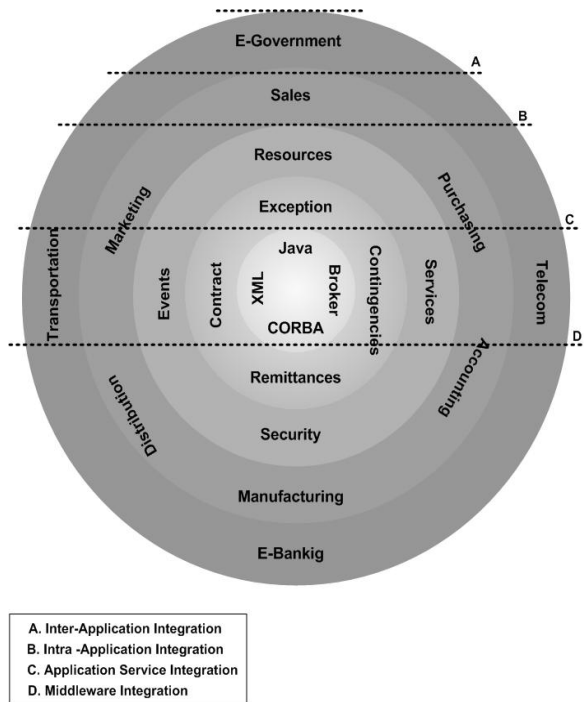
A. Inter-Application Integration
B. Intra -Application Integration
C. Application Service Integration
D. Middleware Integration

Figure 2. Horizontal integration of EAI architecture

3.2. Intra-Application Integration

Intra-application integration seeks to incorporate enterprise applications and exchange information at enterprise level. It requires that enterprise systems leverage and integrate into the new business systems. To attain intra-application integration, this paper proposes to inter-relate the intra-application model through the following integration approaches:

1) Application Service Integration
   a. Data integration
   b. Object integration
   c. Process integration
2) Middleware Integration
   a. Platform integration
   b. Network integration

3.2.1. Application Service Integration. The Application Service Integration is responsible for exchanging data, or functions, from one application to another, to perform joint business functions. This may include integration of custom-to-custom applications, custom to package applications, or package-to-package applications. For the application to be integrated one must first determine which application element; data, object or process, requires integration and what is the access point of the

integration. This paper proposes to integrate the applications through the following approaches.

Data Integration: This integration occurs at the database and data source level and is achieved by migrating data between multiple data sources. Enterprise data are typically distributed among multiple database management systems, and can be resident in relational databases, object-oriented databases, legacy flat files, and many other sources. Each data source may support one, or more, business units and the semantics of these data sources can differ based on the business unit they support. Thus, each data source has different data modelling, different schemas and often run on different platforms. Data integration solutions attempt to move data between multiple data sources by synchronizing the different application semantics that exist in the different data sources, by accessing data from different data sources using a single standard interface [15]. Such integration supports data consistency to all data sources. Various mechanisms exist for moving data between multiple data sources, in both 'real time' and batch modes, such as data replication, extract, transform and load (ETL), data warehouses, data marts and federated databases ([9], [15]).

Object Integration: This integration allows objects, distributed throughout an enterprise, to inter-operate and share functionality. It allows remote object creation and remote object method invocation. To achieve integration of two, or more, applications using object integration, the architect has to rebuild each application using object-integration mechanisms, such as distributed object technology, so that common business logic can be shared, and exposed, for future use by other applications [22]. Object technologies also allow legacy systems to be wrapped and appear as objects. Application wrappers encapsulate data from mainframe user interfaces and transform them into objects. Once the legacy system is wrapped it can participate in an object environment [15]. Application programming interface (APIs) are used to gain access to the objects of packaged solutions [9]. Most packaged applications, such as ERP and SCM, provide APIs to inter-act with underlying applications and business objects. COM, CORBA and J2EE are the most popular distributed object technology tools, for wrapping application logic encapsulated inside the applications [8][15]).

Process Integration: This integration manages the flow of information among the mass of legacy applications, packed applications, e-business applications and other enterprise processes. It defines the dependencies that arise between the

different enterprise objects and processes in order to achieve effective enterprise process. This approach enables the formulation of common business policies, which leads to consistent execution of business processes across the organization. It enables organizations to automate, and integrate, internal, and external, business processes, based on business rules of diverse business systems. Process integration solutions are usually composed of several tools, including a modelling tool, a business process engine and an administration tool [15]. The modelling tool is used to graphically design and model business processes. This tool defines how the information flows between applications by drawing a workflow diagram that links different business logic and resources. The business process engine provides an environment for co-ordinating process execution and process flow to different participants of the process. The administration tool provides administration functionalities such as monitoring the business process and generating reports on the business process performance.

3.2.2. Middleware Integration. Middleware integration is responsible for building a communication infrastructure, for the physical connection and inter-action of all components of the system. This integration addresses the plumbing, the wiring and the run-time environment to accommodate business requirements. It allows executing, sequencing and controlling instances of business processes in conjunction with multi-protocols. Middleware integration can be achieved through network and platform integration.

Platform Integration: This integration allows for integration of a variety of applications running on completely different platforms. Platform integration simply ensures the availability of tools, and processes, to allow applications on different platforms to communicate, both optimally and securely, so that data and functions can be reliably transferred, such as reliably passing data from a NT machine to a Linux machine.

Network Integration: This integration allows the exchange of information across incompatible networks, by translating between dissimilar protocols. Network integration bridges the gap between incompatible networks, so data can be transferred among different systems [22]. The goal of this integration is to achieve connectivity between applications and modules of different networks. Connectivity ensures the data, or message, can be sent from one network to another. Network connectivity is attained through network technologies, and standard protocols such as TCP/IP [22].

4. Case Data

To investigate the effect of the proposed model shown in figure 2, a case study was undertaken in a banking environment. For reasons of confidentiality, the name of the bank cannot be reported and a substitute name Noor will be used instead. Noor is a commercial bank that offers a comprehensive range of financial products and services. The organization consists of 3 international branches, 17 local branches and one subsidiary company. It has up to 3000 employees in 3 countries and over 100,000 customers. In order for the Noor to meet customer specific requirements, they are organized into four units including corporate banking, middle market, commercial banking and government.

Noor was struggling to successfully manage growth. The bank faces considerable challenges in dealing with rapid changes in customer demand, technologies and processes. To stay competitive Noor needed to offer new products and services and to serve new customer and new market. However, The existing Noor infrastructure consists of many incompatible heterogeneous systems causing many problems to the bank. These problems obstruct the bank's efforts to achieve its business objective. The bank struggles to build new application that allow to quickly respond to market opportunity and competitive threats. For example the bank has faced the challenge of integrating the core banking system with the new e-channel applications. The bank has invested huge amounts of money to install the core banking system and most of customer data, loans, deposits and other credit facilities are stored on it. The core banking system is not easy to modify in order to link it with the new channels at the same time discarding the core banking system is never an easy choice

Thus Noor needed a new application that can reuse business components from it core kit. Noor realise that a robust and flexible infrastructure will help it reliably leverage the Web to deliver the new application to the bank growing customer base. The infrastructure needed to provide high performance, reliability and security 24 hours a day and 7 days a week.

In addressing these problems, the Chief Executive Office (CEO) asked IT director to find a solution for introducing new e-delivery channels and integrating the existing IT infrastructure to achieve bank competitive goals. The IT director asked the IT development manager to introduce solutions for the

above problems. The IT development manager after studying number of technologies, was convinced that an EAI solution would allow integration of inter and intra bank applications in a more flexible and maintainable way. The EAI solutions allow the bank to link its core banking legacy system to new e-delivery channels and achieved interoperability and portability across various bank functions. However, the task of implementing EAI solution is a complex one. According to the IT development manager the most important barriers to application integration is understanding the high complexity of business processes. For this purpose an enterprise architecture model is required to formulate the enterprise strategy and show the relationships between the disparate functional areas. The IT development manager believes that the existence of proper architecture model will avoid functionality overlaps, avoids the gaps and reduce integration cost. Thus, the bank IT department took a decision to develop architecture model before proceeding to the implementation of EAI solution. Due to confidentiality reason, the bank is reluctant to share their EAI architecture. They believe that sharing their EAI architecture with others will lead to loss of a competitive advantage. However, the IT development manager reported informally that the diversity of the enterprise application integration approaches has made selection of an EAI solution difficult and therefore there is a need for architecture to understand business process view before identifying appropriate information infrastructure that support these processes. It appears that Noor had difficulty in finding the most suitable EAI solution due to diversity EAI solutions. It seems that the existence of an architecture model for enterprise application integration support the organization to understand their business need and identify appropriate information infrastructure that support the business processes.

IT development manager and e-business manager were asked to comment on the proposed architecture model by the authors for enterprise application integration. The IT development manager believes that the propose model will reduce intra and inter application integration cost as well as data and functional redundancies cost. The IT development manager indicates that this model will allow the services to be flexibly assigned to the proper technological channel and common functionalities can be reused more efficiently. The e-business manager commented that the proposed model allow to drive the integration needs from business imperative rather than product capabilities and this will allow to capture business integration needs and

close the gaps between complex processes and the IT implementation. The e-business manager emphasised that this model can bring a significant cost advantage through the reuse and reducing time required to implement. This model also facilitates rapid realization of business integration goals and it surpasses the middleware to provide structured solutions for application interoperability. E-business manager suggested addition to the model to be scalable enough to handle unpredictable transaction volume in a constantly changing environment. The authors added the transaction service model to models contractual obligations, exceptions, contingencies, remittances and contract violations.

5. Conclusion

This paper has achieved two objectives. First, it has provided a comparison of existing enterprise application integration approaches and classified them into two general categories. In doing so, the paper has united different EAI concepts and terminology and provided a deeper understanding of enterprise application integration approaches.

Second, the paper has proposed a new EAI architecture that captures different integration aspects into models and structures them into inter-related layers. Building on the idea of OMG's Model Driven Architecture, this architecture aims at achieving a clear separation of concerns and enabling the business aspect of an application and its technological aspect to evolve independently of each other. We believe that the development of this new architecture is an important contribution to EAI.

An interpretive, qualitative case study on commercial bank in kingdom of Bahrain was conducted for this research. The empirical data demonstrates that the existence of an architecture model for enterprise application integration plays a significant role in adopting EAI solution. It appears from the empirical data that developing an architecture model that can understand both the business and technological concept for adopting EAI solution result in achieving competitive advantage. Empirical evidence further suggests consideration of unpredictable transaction in the architecture. Consequently, the authors have revised the proposed model and have now added the transaction service model to models contractual obligations, exceptions, contingencies, remittances and contract violations.

Our next step is to investigate a software tool for specifying our architecture and implementation technologies to support the architecture.

6. References

[1] Brown, L., *Integration Models: Templates for Business Transformation,* SAMS Publishing, USA, 2000.

[2] Chalmeta, R., Campos, C. and Grangel, R., "References Architectures for Enterprise Integration", *Journal of Systems and Software*, Vol. 57, No.3, 2001, pp. 175-191.

[3] Duke, S., Makey, P. and Kiras, N., *Application Integration Management Guide: Strategies and Technologies*, Butler Group Limited, Hull, UK. 1999.

[4] Giachetti, R., "A Framework to Review the Information Integration of the Enterprise", *International Journal of Production Research*, Vol. 42, No.6, 2004, pp 1147-1166.

[5] Grimson, J., Grimson, W. and Hasselbring, W., "The SI Challenge in the Health Care", *Communications of the ACM,* Vol.43, No.6, 2000, pp. 49-54.

[6] Hasselbring, W., "Information System Integration", *Communications of the ACM*, Vol.43, No.6, 2000, pp. 33-38.

[7] Lam, W. and Shankararaman, V., "An Enterprise Integration Methodology", *IT Professional,* Vol.6, No.2, March-April 2000, pp. 40-48.

[8] Linthicum, D., *Enterprise Application Integration*, Addison-Wesley, Massachusetts, USA, 1999.

[9] Linthicum, D., *B2B Application Integration*, Addison-Wesley, Massachusetts, USA, 2001.

[10] Maheswari, P., "Enterprise Application Integration using a Component based Architecture", Proc. 27th Annual International Computer Software and Applications Conference, 2003, p 557-562

[11] MDA Guide Version 1.0.1, Document Number: omg/2003-06-01, 12th June 2003. Available at www.omg.com.

[12] Pushmann, T. and Alt, R., "Enterprise Application integration: the case of the Robert Bosch Group", Proc. 34th Hawaii International Conference on System Sciences (HICSS'01), Hawaii, USA, 2001.

[13] Ring, K. and Ward-Dutton, N., *Enterprise Application Integration: Making the Right Connections,* Ovum Ltd, London, UK, 1999.

[14] Ruh, W., Maginnis, F., and Brown, W., *Enterprise Application Integration: A Wiley Tech Brief,* John Wiley &Sons Inc., New York, USA, 2000.

[15] Samtani, G., Healey, M. and Samtani, S., *B2B Integration: A Practical Guide to Collaborative E-commerce,* Imperial College Press, London, UK, 2002.

[16] Sharma, R., Starns, B. and Ng, T., *J2EE Connector Architecture and Enterprise Application Integration*, Addison-Wesley, Mass., USA, 2001.

[17] Themistocleous, M. and Irani, Z., "Towards a Novel Framework for assessment of Enterprise Application Integration Packages", Proc. 36th Hawaii International Conference on System Sciences (HICSS'03), Hawaii, USA , 2003, pp.10.

[18] Themistocleous, M., and Irani, Z., "Benchmarking the Benefits and Barriers of Application Integration", *Benchmarking: An International Journal*, 2001, Vol.8,No. 4, pp.317-31.

[19] Themistocleous, M., "Justifying the Decisions for EAI Implementations: a Validated Proposition of Influential Factors", *Journal of Enterprise Information Management,* 2004, Vol. 17,No. 2, pp.85-104.

[20] Vasconcelos, A., Silva, M., Fernandes, A. and Tribolet, J., "An Information System Architectural Framework for Enterprise Application Integration", Proc. Hawaii International Conference on System Sciences (HICSS), Vol. 37,2004 , pp.3541-3549.

[21] Yang, J. and Papazoglou, M., "Interoperation Support for Electronic Business", *Communications of the ACM,* Vol. 43, No.6, 2000, p 39-47.

[22] Zahavi, R., *Enterprise Application Integration with CORBA,* John Wiley and Sons Inc, New York, USA, 1999.