EVALUATING THE ADOPTION OF ENTERPRISE APPLICATION INTEGRATION IN MULTINATIONAL ORGANISATIONS

A thesis submitted for the degree of Doctor of Philosophy by

Marinos G. Themistocleous

Department of Information Systems and Computing, Brunel University

March 2002

BRUNEL UNIVERSITY

PUBLICATION OF THESES

Please complete in block capitals

NAME: MARINOS THEMISTOCLEOUS

DEPARTMENT: DISC

TITLE OF THESIS: EVALUATING THE ADOPTION OF ENTERPRISE APPLICATION INTEGRATION IN MULTINATIONAL ORGANISATIONS

PLEASE COMPLETE EITHER PART A OR PART B

A. I agree that the abstract of my thesis may be published by the University without further reference to me.

In accordance with the University's Handbook of Procedures, the Head of Library Services may allow my thesis to be copied in whole or in part without further reference to me. Such Authority shall apply only to single copies made for study purposes and shall be subject to normal conditions of acknowledgement.

(See notes overleaf)

Signature: [signature]

Date: 22/04/02

If you have completed Part A please return this form direct to the Library with evidence of your award.

B. I request that my thesis be held under confidential cover in the Library for a period of _______ years for the following reasons:

____

Once the period of confidentiality has expired, I agree that the conditions set down in Part A will apply without further reference to me.

Signature: ____________________

Date: ____________________

Please pass this form to your supervisor who should sign below indicating his/her agreement to the retention of the thesis under confidential cover.

Signature of Supervisor: ____________________

Additional comments: ____________________

Signature ____________________
(Senior Assistant Registrar)

Part B, when complete, should be returned to the Senior Assistant Registrar (Records & Systems) in the Registry.

PhD Abstract

A review of normative literature, in the field of Information Systems (IS) integration, indicates that traditional approaches to applications integration have failed to result in flexible and maintainable IT infrastructures. In addressing this issue, a new technology called Enterprise Application Integration (EAI) has emerged and addresses most of integration problems by resulting in the development of reusable and manageable IT infrastructures. Enterprise application integration is a new research area with many research issues needing to be investigated. At this end, EAI adoption has not efficiently studied with organisations and researchers needing to understand and analyse EAI adoption.

This work examines the introduction of enterprise application integration in multinational organisations and proposes a novel model for its adoption. The model is based on a comprehensive set of factors that influence the introduction of EAI in organisations. Since there is an absence of theoretical models for EAI adoption, the proposed model adapts factors that influence the adoption of other integration technologies such as Electronic Data Interchange (EDI). Additional factors like an evaluation framework that supports decision-making have been considered by the author as factors that influence EAI adoption.

In moving from the conceptual to the empirical, the work is based on a qualitative case study approach to examine the concepts of the proposed model for the adoption of EAI. In doing so, two case studies were conducted at multinational organisations and presented and analysed. However, during the empirical research complementary factors also emerged, which resulted in modifications being made to the previously presented conceptual model. In interpreting from empirical data, it appears that ten main factors influence the adoption of EAI namely: (a) benefits; (b) barriers; (c) costs; (d) internal pressures; (e) external pressures; (f) IT infrastructure; (g) IT sophistication; (h) an evaluation framework for the assessment of integration technologies; (i) evaluation framework for the assessment of EAI packages and, (j) support.

The proposed model makes novel contribution at two levels. First, at the conceptual level, as it incorporates factors identified separately in previous studies as influencing adoption of other integration technologies. These factors are used for the development of a consistent model for the adoption and evaluation of EAI. Secondly, the concepts of the proposed model can be used for the adoption of inter-organisational information systems. The proposed model can be used as a decision-making tool to support management when taking decisions regarding the adoption of EAI. Additionally, it can be used by researchers to analyse and understand the adoption of application integration.

Acknowledgements

I would like to thank my supervisors, Dr. Zahir Irani and Prof. Bob. O'Keefe, of the department of Information Systems and Computing (DISC) at Brunel University, for their time, guidance and kind support during my thesis. Also, many thanks to the DISC for internally funding my doctoral studies through, a 'full' scholarship.

I acknowledge the case studies companies and their staff, for their time and assistance in collecting empirical data to ensure the completion of this thesis. Many thanks to Thomas Puchamann and Dr. Alt Rainer of University of St. Gallen (Switzerland) as well as to Prof. Mike Papazoglou and Dr. Willem-Jan van den Heuvel of University of Tilburg (Holland) for their support during this research.

I would like to extend my thanks to Prof. Ray Paul, Prof. Bob O'Keefe and Prof. Rob Macredie to offer me a lectureship, before the completion of my thesis. Many thanks to Prof. Georgios Doukidis of Athens University of Economics and Business (Greece) for motivating me into becoming a Doctor of Philosophy (PhD) and joining Brunel University. Last, but not least, I would like to thank my colleagues at the attic and my flatmates for their encouragement and support.

Στην οικογένεια μου
για την συμπαράσταση και τη δύναμη
που μου έδινε όλα αυτά τα χρόνια

Στη Νάντια
για την ύπαρξη της, την αγάπη της και την αφοσίωση
που μου παρείχε τα τελευταία οκτώ χρόνια τη ζωής μου

Declaration

This thesis gives an account of the research undertaken by the author. Some of the material contained herein has been presented in the form of the following publications:

Refereed Journal Papers *Published/Accepted*

Themistocleous, M. and Irani, Z. 2002a. 'Evaluating Enterprise Application Integration Technologies: A Novel Frame of References', *European Journal of Operational Research,* In Press.

Themistocleous, M. and Irani, Z. 2002b. 'Novel Taxonomy for Application Integration', *Benchmarking: An International Journal,* **9**(2): In press.

Themistocleous, M., Irani, Z. and O'Keefe, R. 2001. 'ERP and Application Integration: Exploratory Survey', *Business Process Management Journal,* **7**(3): 195-204.

Themistocleous, M. and Irani, Z. 2001a. 'Benchmarking the Benefits and Barriers of Application Integration', *Benchmarking: An International Journal,* **8**(4): 317-331.

Refereed Journal Papers *Under Review*

Themistocleous, M. and Irani, Z. 2002c. 'Application Integration and Supply Chain Management', *European Journal of Operational Research,* Under Review

Professional Conference Papers *Published/Accepted*

Themistocleous, M. and Irani, Z. 2002d. *'Evaluating and Adopting Application Integration: The Case of a Multinational Petroleum Company'*, Proceedings of Thirty-Fifth

Annual Hawaii International Conference on System Sciences, Hicss 35, Big Island, Hawaii, USA, [CD Proceedings].

Themistocleous, M. and Irani, Z. 2002e. *'A Model for Enterprise Application Integration Adoption and Evaluation'*, Proceedings of Second International Conference on Systems Thinking in Management (ICSTM-2002), University of Salford, Manchester, UK, Forthcoming.

Themistocleous, M., Irani, Z. and Love, P. 2002. *'Enterprise Application Integration: An Emerging Technology for Integrating ERP and Supply Chains'*, Proceedings of Tenth European Conference on Information Systems (ECIS 2002), Gdansk, Polland, Forthcoming.

Themistocleous, M. and Irani, Z. 2001b. *'Evaluating Application Integration: An Exploratory Case Study'*, Proceedings of Seventh Americas Conference on Information Systems, AMCIS 2001, Boston, Massachusetts, USA, pp. 1376-1380.

Themistocleous, M., Irani, Z., O'Keefe, R. and Paul, R. 2001. *'ERP Problems and Application Integration Issues: An Empirical Survey'*, Proceedings of Thirty-Four Annual Hawaii International Conference on System Sciences, Hicss 34, Maui, Hawaii, USA, [CD Proceedings].

Themistocleous, M., Irani, Z., Psannis, K. and Vrehopoulos, A. 2001. *'Application Integration of Information Technology: Classification of Benefits and Barriers'*, Proceedings of International Workshop on New Models of Business: Managerial Aspects and Enabling Technology, St.Petersburg, Russia, pp. 153-161.

Themistocleous, M. and Irani, Z. 2000. *'Taxonomy and Factors for Information System Application Integration'*, Proceedings of Sixth Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 955-959.

Themistocleous, M., Irani, Z. and Sharif, A. 2000. *'Evaluating Application Integration'*, Proceedings of 7th European Conference on Evaluation of Information Technology (ECITE 2000), Dublin, Ireland, pp. 193-202.

Professional Conference Papers *Under Review*

Love, P., Irani, Z., Burn, J. and **Themistocleous, M.** 2002. *'Constructing Strategies for e-Commerce: the Role of e-commerce in Small Construction Companies'*, Proceedings of Eighth Americas Conference on Information Systems, AMCIS 2002, Dallas, Texas, USA.

Themistocleous, M. and Irani, Z. 2002f. *'A Model for Adopting Enterprise Application Integration Technology'*, Proceedings of Working Conference on the Adoption and Diffusion of IT in an Environment of Critical Change (IFIP WG8.6), Sydney, Australia.

Themistocleous, M. and Irani, Z. 2002g. *'Enterprise Application Integration Adoption and Evaluation'*, Proceedings of Eighth Americas Conference on Information Systems, AMCIS 2002, Dallas, Texas, USA.

Teaching Text Book

Doukidis, G., **Themistocleous, M.**, Drakos, W. and Papazafeiropoulou, A. 1998. *'Electronic Commerce'*, New Technologies Publications, Athens, Greece.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Summary

During the last decades, enterprises have focused on Information Technology (IT) and implemented various applications to automate their business processes. These applications were not developed in a co-ordinated way but have evolved as a result of the latest technological innovation. The IT infrastructure in several organisations consists of autonomous and in many cases heterogeneous solutions. This situation has caused various integration problems as applications could not co-operate and thus, disparate IT solutions could not be bridged together.

In recent years, an emerging category of integration software called Enterprise Application Integration (EAI) or simply application integration has attempted to effectively address many integration problems and thus, result in the development of flexible, and maintainable integrated information systems. It is achieved through the incorporation of functionality from a diversity of systems. Application integration is a new research area and therefore, scientific research and literature around it, remains limited. Yet, the impact of application integration on organisations remains under explored and reported. Enterprises seek answers to the impact of integration, as it will help them realising the benefits, the barriers, the risks, the costs and changes that are associated with the adoption of application integration.

In addressing this issue, the research presented in this dissertation investigates and evaluates *the impact of application integration on organisations as well as its adoption.* This chapter explains why existing information systems have failed to provide solutions to integration problems and discusses the need for the development of a single enterprise infrastructure. The aim and the objectives of this research are defined, with an outline of the dissertation presented at the end of this chapter.

1.1 Background to the Research Problem: Information Systems Evolution and the Need for Integration

The need for integration is not new but, it existed since applications moved from central processors to distributed systems and networks. This need has emerged as disparate Information Systems (IS) that automate business processes have run on different computer platforms and have been based on a diversity of standards, operating systems and computer languages. In the past, organisations rarely had a single approach for implementing information systems and developed applications without common enterprise architectural planning (Markus and Tanis, 1999; Brown, 2000). Many information systems were not developed to incorporate with other solutions but rather, focused on solving point problems and usually form autonomous *islands of technology* (Swenson and Cassidy, 1993; Duke *et al.*, 1999).

The number of incompatible islands of technology has increased rapidly, with organisations seeking ways to integrate these systems. The reason for this was that islands of technology have a number of drawbacks, which affect organisations. For instance, each individual application needed to store and handle its own data, since applications do not share data or services. In doing so, this resulted in increased redundancy of data and systems functionality. This redundancy generates serious data integrity problems, as data that deal with a specific object (e.g. a customer) were not updated similarly among applications. Such integrity problems have caused functional problems since managers or accountants could not have a clear view regarding the data and therefore, could not analyse them.

Another serious drawback of islands of technology was their high operational cost. Since each application was based on a different platform or operating system, organisations need more experts to support and maintain all these applications. Additionally, as business processes were partially automated and information systems were not integrated, many tasks require manual intervention, which also increases operational cost. To overcome all these problems, organisations started seeking ways to integrate their applications.

Initially, enterprises attempted to address integration by interconnecting their disparate applications but the number of interconnections required increased rapidly, as in many cases each application had to be interconnected with all the others. Themistocleous *et al.*(2001b) estimate that for **x** applications a total of

$$x*(x-1)/2$$

interconnections are needed when each application is interconnected with the rest applications.

However, interconnecting applications is not an easy task, since the constrains and requirements of existing software solutions should be taken into consideration to piece applications together. There is also a complexity of existing IS which, in many cases have fixed and rigid structures for messages, interfaces and databases. Additionally, there is a lack of common definitions, structures, business concepts and standards, which makes interconnectivity more complex, as a diversity of tools are required to interconnect information systems (Robertson, 1997; O'Callaghan, 1999). The reason for this is that applications do not based on a common programming language or platform. Moreover, there is a lack of documentation, especially in legacy[1] systems, and often important technical information is missed (Brodie and Stonebraker, 1995; Lloyd *et al.*, 1999).

To achieve interconnectivity among systems, programmers map data from source's application format to target's since applications require compatible data to store and manipulate them. In support of this, programmers invade and alter the code of systems in order to map data and automate these interconnections. Therefore, the maintenance of these interconnected IT solutions becomes a serious issue for concern, as changes in one system often required the altering of all interconnected applications (Brodie and Stonebraker, 1995; Butler Group, 1998; Brown, 2000). As a result, interconnection has proved a complex, cost consuming, non-flexible and non-manageable solution.

[1] Legacy systems are large software systems, vital for organisations, which significantly resist modification and evolution to meet new business requirements Bennett, K. 1995. 'Legacy systems: coping with success', *IEEE Software,* **12**(1): 19-23, Brodie, M. and Stonebraker, M. 1995. *'Migrating legacy systems',* Morgan Kaufmann Publishers, San Francisco, USA, 1558603301.

1.2 The Limited Scope and Success of Enterprise Resource Planning Systems

During the 1990s, Enterprise Resource Planning (ERP) technology emerged as an approach to integration problem (Davenport, 1998). ERP systems do not integrate disparate applications but, replace the need to integrate. According to Gibson *et al.* (1999) ERP systems are integrated software packages that automate core corporate activities such as finance, human resources, manufacturing and supply and distribution. Enterprise Resource Planning systems let a company share common data and practices across the enterprise, and produce and access information in a real-time environment. These systems are designed to solve the fragmentation of information in large business organisations, and integrate all information flows within a company.

Chung and Snyder (2000) claim that ERP systems support common processes and best practices with many organisations attempting to parameterise ERP packages to better support their business processes and strategy. However, customisation is a difficult task that causes serious integration problems, as ERP systems are complex, non-flexible and often not designed to collaborate with other autonomous applications (Sumner, 1999).

Linthicum (1999) and Zahavi (1999) characterise ERP systems as monolithic solutions that are not designed to co-operate with other applications. As a result, enterprise integration can be achieved when organisations abandon existing applications and develop a complete ERP solution. Therefore, the more ERP modules adopted, the more incorporation is achieved. Nevertheless, Makey (1998), Markus and Tanis (1999) and Themistocleous *et al.* (2001a) indicate that companies often do not adopt *all* ERP modules but a subset of them. The reasons are many including: (a) enterprises use existing systems alongside ERPs; (b) ERP modules cost considerable amounts of money and, (c) there is a lack of time or justification to replace existing systems with new ERP modules. Even in cases where organisations purchase all ERP modules from a single vendor, ERP packages can not automate more than 30% of company's application (Seeley, 1999; Stefanou, 2000). In contrast, Makey (1998), Holland and Light (1999) and Kelly *et al.* (1999) report that ERP systems cover up to 70-80% of a company IT requirement. Regardless, organisations do not abandon all their existing systems when adopting ERP packages, several applications (e.g. legacy systems) often co-exist alongside enterprise systems (Puschmann and Alt, 2001).

The amount of legacy systems in use remains high (Lloyd *et al.*, 1999) as they provide reliable solutions. In support of this, Themistocleous *et al.* (2001a) report that 38% of companies do not replace their legacy systems when adopting Enterprise Resource Planning (ERP) solutions. In addition, Ring and Ward-Dutton (1999) suggests there is often no time to replace legacy systems, Ruh *et al.* (2000) explaining that replacement is a high risk process. O'Callaghan (1999) supports the claim that the replacement of legacy systems is too expensive with Brodie and Stonebraker (1995) explaining that it takes too long to realise the benefits. Apart from the incorporation of existing systems, organisations have to integrate new applications (e.g. e-business solutions, supply chain applications) with ERP package.

All observations discussed in this section indicate that ERP systems can be considered as a *partial* solution to enterprise integration as other applications co-exist along-side ERP packages. As a result, Loos (2000), Meier *et al.* (2000) and Schonefeld and Vering, (2000) indicate that *there is a need to integrate enterprise systems with new or existing applications.* Although, ERP systems were not designed to incorporate other autonomous applications (Schonefeld and Vering, 2000), a diversity of approaches, techniques and tools can be used to achieve integration between ERP systems and disparate applications (Loos, 2000; Meier *et al.*, 2000; Morgenthal and La Forge, 2000). Integration can be achieved at a data, object, and interface level (Brown, 2000; Ruh *et al.*, 2000; Schonefeld and Vering, 2000), with *enterprise application integration* supporting all these types integration levels.

1.3 The Important Role of Enterprise Application Integration in e-business Era

The tremendous adoption of *Electronic Commerce* (e-commerce) applications during the last decade has amplified the need for integration, since these systems have to be integrated with existing applications. e-commerce refers to conducting business electronically using computers and networks, and focuses on the integration and automation of business processes (Kalakota and Robinson, 1999). Doukidis *et al.* (1998) suggests e-commerce provides access to global markets through the Internet, and can lead to competitive advantages as it improves sales channels, simplifies and automates transactions. In addition, it achieves cost reduction and user satisfaction, and improves relationships with customers and suppliers. The advantages that e-commerce offers can be transacted in the redesign of business practices, strategies and models (Timmers, 1998). The expansion of e-commerce applications has resulted in revenues of billion of dollars per year for those companies involved.

In recent years, a number of e-commerce enabled companies have failed (e.g. http://www.e-toy.com) with Hooft and Stegwee (2001) reporting a lack of management support, insufficient budgets and cultural issues as the main reasons for this failure. Bhatt and Emdad (2001) and Kalakota (2000) support that *integration* plays a critical role for the success of e-commerce applications since it allows system co-operation, and supports real-time transactions. Morgenthal and La Forge (2000) among others report that e-commerce applications have to be incorporated with back-office systems to automate and integrate business processes and support real-time transactions. The reason for this is that e-commerce applications are not integrated solutions. As a result, there is a need to piece together e-commerce solutions with existing IT infrastructures to allow them to function in an integrated way thus, allowing enterprises gain e-commerce advantages (Timmers, 1998).

The need for integration has led Kalakota and Robinson (1999) among others to separate e-commerce solutions into integrated and non-integrated applications. Based on this categorisation, the term *e-business* was introduced to describe integrated e-commerce applications with Kalakota and Robinson (1999) suggesting that e-business solutions achieve structural transformation and are based on flexible and manageable architectures. Likewise, Isakowitz *et al.* (1998) and Pant and Ravichandran, (2001) claim that e-business systems allow transactions to be conducted in an integrated way by removing constrains imposed by diverse systems. Another explanation of the term e-business was given by Linthicum (2000) who reports that it refers to Business-to-Business (B2B) applications where e-commerce to Business-to-Consumer (B2C) applications. In the context of this dissertation, the term *e-business* is adopted to refer to IS that allow organisations to do business electronically using computer networks and it refers to both B2C and B2B solutions.

The adoption of e-business solutions has increased the need for integrated inter-organisational applications to achieve competitive advantages (Kalakota and Robinson, 1999). In this context, Supply Chain Management (SCM) needs to be integrated with intra and inter-organisational systems, as it shares critical information between the partners of a supply chain (e.g. suppliers, distributors, retailers etc.). However, the integration of transactions along a supply chain is a difficult task as it requires a thorough understanding of organisational integration across the supply chain and leverages various sources of expertise (Vankatraman and Henderson, 1998). Diverse categories of IS (e.g. custom, packaged etc) are used to monitor and co-ordinate a supply chain. These applications were usually not designed to collaborate with other systems, as partners have developed their applications independently and without any co-ordination (Kalakota and Robinson, 1999). Therefore, supply chain integration is related to the incorporation of all involved custom systems, packaged solutions and e-business applications into a unified infrastructure (Linthicum, 1999). In addition, integration is becoming more difficult as the issue of *timing* is critical for the efficient operation of all systems, and the majority of e-business and supply chain applications require real-time incorporation (Jayaram *et al.*, 2000).

Emmelhainz (1993) reports that many companies have used Electronic Data Interchange (EDI) technology and Value Added Networks (VAN) to exchange their business documents in an integrated way, and piece together their supply chains. Although, EDI achieves data integration it is not adequate for enterprise and cross enterprise incorporation, as it has a number of drawbacks (Choudhury, 1997; Kim and Umanath, 1999). EDI is a complex and invasive technology that does not achieve process integration and does not provide the flexibility and the maintainability demanded (Nissen, 2000). The complexity and high cost of EDI as well as the emergence of the Internet as a global platform for e-business has led organisations to adopt open standards such as Extensible Markup Language (XML) to facilitate their transactions and achieve integration. Linthicum (2000) claims that although XML supports the integration of Internet based transactions, it can not address all integration problems, as many transactions are not running over the Internet but on back-office systems. In addition, organisations consist of a set of complex incompatible information systems with diverse information formats, heterogeneous computing platforms and various programming models that require technologies to piece together all these systems. As a result, organisations have started working with more advanced technologies that is provided by application integration to efficiently incorporate their supply chains (Kalakota, 2000).

1.4 Enterprise Application Integration

A diversity of terms such as Enterprise Application Integration (EAI) (Brown, 2000), Application Integration (AI) (Sprott, 2000), Systems Integration (SI) (Hasselbring, 2000), Value Chain Integration (VCI) (Yang and Papazoglou, 2000), Supply Chain Integration (SCI) (Linthicum, 1999), Extended Business Integration (EBI) (Markus, 2000) and e-business Integration (e-business I) (Linthicum, 2000) were presented in the literature to define the information system integration area. However, in attempting to navigate through this confusion, Themistocleous *et al.*(2000) distinguished much of this terminology. In the context of this dissertation, the terms *Enterprise Application Integration* (EAI) and *Application Integration* (AI) are used equally to refer to the integration area. A summary of definitions on integration area are presented and evaluated in Appendix A. Enterprise application integration is defined as the:

> *"unrestricted sharing of information between two or more enterprise applications. A set of technologies that allow the movement and exchange of information between different applications and business processes within and between organisations."*
>
> *Linthicum (1999, p.354)*

EAI is a new generation of integration software that incorporates functionality from disparate systems and leads to flexible and maintainable solutions (Zahavi, 1999). It addresses more effectively the need to integrate both intra and inter-organisational systems by incorporating functionality from disparate applications and thus, maximise their benefits from the use of e-commerce and e-business applications. EAI combines traditional integration technologies (e.g. database-oriented middleware, interface-based technologies, distributed object technologies, etc.) with new application integration technologies (e.g. message brokers) to support the efficient incorporation of IS. Application integration results in supporting data, objects and processes incorporation as well as custom applications, packaged systems and e-business solutions integration (see Figure 3.2).

1.5 Research Aim and Objectives

1.5.1 Research Aim

The research reported in this dissertation, is based on the rationale that enterprise information systems do not provide an integrated IT infrastructure. Instead, they co-exist alongside disparate applications (Zahavi, 1999; Grimson *et al.*, 2000; Sprott, 2000). The reason for this is that enterprise systems do not cover all the IT requirements of an enterprise (Makey, 1998; Holland and Light, 1999; Kelly *et al.*, 1999; Seeley, 1999). Enterprise systems are often not designed to incorporate other internal or external autonomous applications (Schonefeld and Vering, 2000). Moreover, there is a need to integrate enterprise systems with other applications (Loos, 2000; Meier *et al.*, 2000; Schonefeld and Vering, 2000) as a non integrated IT infrastructure adds more complexity to organisations and costs more (operational, maintenance, functional, management costs, etc.). Traditional approaches to enterprise application integration tend to be techno-centric and address integration issues by solving point problems and interconnecting systems rather than developing an integrated and manageable IT infrastructure (Linthicum, 1999). EAI technology is a new class of integration software that addresses the need for intra-organisational and inter-organisational application incorporation. EAI leads to the development of flexible and maintainable IS by securely incorporating functionality from disparate applications. Unlike, other integration technologies such as Electronic Data Interchange (EDI), EAI attempts to fully automate and integrate business processes. In doing so, EAI uses a set of technologies like message brokers and workflow tools to support process integration. In addition, integration is achieved in more advanced and flexible ways when EAI is used (e.g. point-to-point interconnections are eliminated).

However, various limitations restrict the widespread of EAI. Firstly, there is a plethora of EAI products that solve integration problems, yet none of these address all the different types (e.g. package to package integration, custom applications integration, data integration, components integration etc) of integration requirements (Klasell and Dudgeon, 1998; Ring and Ward-Dutton, 1999). Another serious limitation is that the impact of EAI on organisations and its adoption has not been comprehensively studied and analysed.

Therefore, it is not clear which business parameters are affected by EAI and to what extend those factors influence the decisions for its adoption.

Therefore, to better understand the issues surrounding EAI, organisations may be benefited from a *frame of references* to support the integration of applications. Such a frame of references will better help organisations to understand the impact of EAI on business performance and structure, before proceeding with their investment strategy. The proposed frame of references may help enterprises support effective management by eliminating the cost and risk of their EAI investment. In doing so, maximising business benefits, gaining strategic advantages, and transforming the organisation. As a result, the aim of this dissertation is to:

> *Evaluate the adoption of Application Integration on multi-national enterprises. In doing so, resulting in the development of a frame of references that translates into a model that can be used to support decision-making.*

1.5.2 Research Objectives

To reflect upon the research aim, the research objectives of the study are made clear:

- To conduct a comprehensive literature review in the area of enterprise application integration with a particular focus on EAI adoption and evaluation.
- To identify barriers, benefits and costs associated with the adoption of application integration.
- To assess approaches associated with the adoption of application integration. In doing so, identifying *why, how* and in *what* way application integration has been adopted.
- To identify and critically evaluate those technologies associated with application integration. In support of this, identify *why* a particular technology was adopted and establish its scope.

- To develop and propose a frame of references that can be translated into a model for EAI adoption and evaluation.

1.6 Dissertation Outline

The structure of this PhD dissertation follows the methodology described by Phillips and Pugh (1994) and consists of four elements namely: (a) background theory; (b) focal theory; (c) data theory and (d) novel contribution. Background theory focuses on assessing the field of research and identifying the problem domain (see Chapter 2). The second element of the dissertation (focal theory) deals with generating conceptual models. This is explained and discussed in Chapter 3. Data theory addresses issues such as: (a) the most appropriate epistemological stance to adopt; (b) the development of a suitable research methodology and, (c) the conditions affecting the choice of research strategy. These issues are discussed in Chapter 4 of this dissertation. In addition data theory deals with the data collection process and analysis, which is reported, in Chapter 5. The fourth element (novel contribution) is concerned with aligning the importance of the thesis, to the development of the discipline being researched (see Chapters 6). The dissertation is composed of seven chapters with each of the chapters providing an understanding to various issues viewed to be critical for this research. The dissertation outline is illustrated at Figure 1.1 and is explained in the following paragraphs.

Chapter 1: Introduction

Chapter 1 begins by providing an introduction to the main issues that the research will address. These issues focus on the need to integrate intra-organisational and inter-organisational applications in a more flexible and maintainable way. Thereafter, the aim and objectives of the research were stated. The chapter ends with the dissertation outline.

Chapter 2: Literature Review – Background Theory

Having provided a brief introduction to the area of research and establish the scope, the dissertation then begins to review the literature on EAI. Initially, the existing models for adopting integrated technologies are discussed. Thereafter, Chapter 2 investigates the nature of application integration and proposes a novel taxonomy for categorising EAI types. The dissertation then identifies and defines classifications of information system types that are integrated. Chapter 2 ends by discussing and classifying EAI benefits, EAI barriers, EAI costs and external pressures.

Chapter 3: Conceptual Model for EAI Adoption and Evaluation – Focal Theory

As reported in Chapter 2, one important barrier to application integration is the selection of an appropriate set of integration technologies suitable when adopting EAI. Chapter 3 attempts to overcome this barrier by reviewing the diversity of integration technologies and evaluating them by proposing a novel evaluation framework. The evaluation framework contributes towards a better understanding of the capabilities of each technology, and highlights possible combinations of integration solutions. Thereafter, a novel model for the adoption of EAI is developed and analysed. The model proposes that eight factors influence the adoption of EAI namely: (a) EAI costs; (b) EAI benefits; (c) EAI barriers; (d) external pressures; (e) support; (f) the level of IT sophistication; (g) the limitations of existing IT infrastructure and, (h) an evaluation framework that supports organisations to assess integration technologies. The proposed model makes a novel contribution at two levels. First, at the conceptual level, as it incorporates factors identified separately in previous studies as influencing the adoption of integration technologies such as EDI. These factors are used for the development of a consistent model for the adoption of application integration. Secondly, the concepts of the proposed model can be used for the adoption of inter-organisational information systems. The proposed model can be used as a decision-making tool and thus, support management when taking decisions regarding the adoption of EAI. Additionally, the model can be used by researchers to analyse and understand the adoption of EAI.

Chapter 4: Research Methodology – Data Theory

Chapters 2 and 3 are setting the background of this research and have helped the author to understand and identify research issues. To undertake the research that focuses on these issues, a research methodology has to be followed. The reasoning behind the research methods is stated within chapter 4. The inherent problems within the various research philosophies are stated and the suitability to this research is provided. The research strategies existing within the IS field are also described and discussed within this chapter.

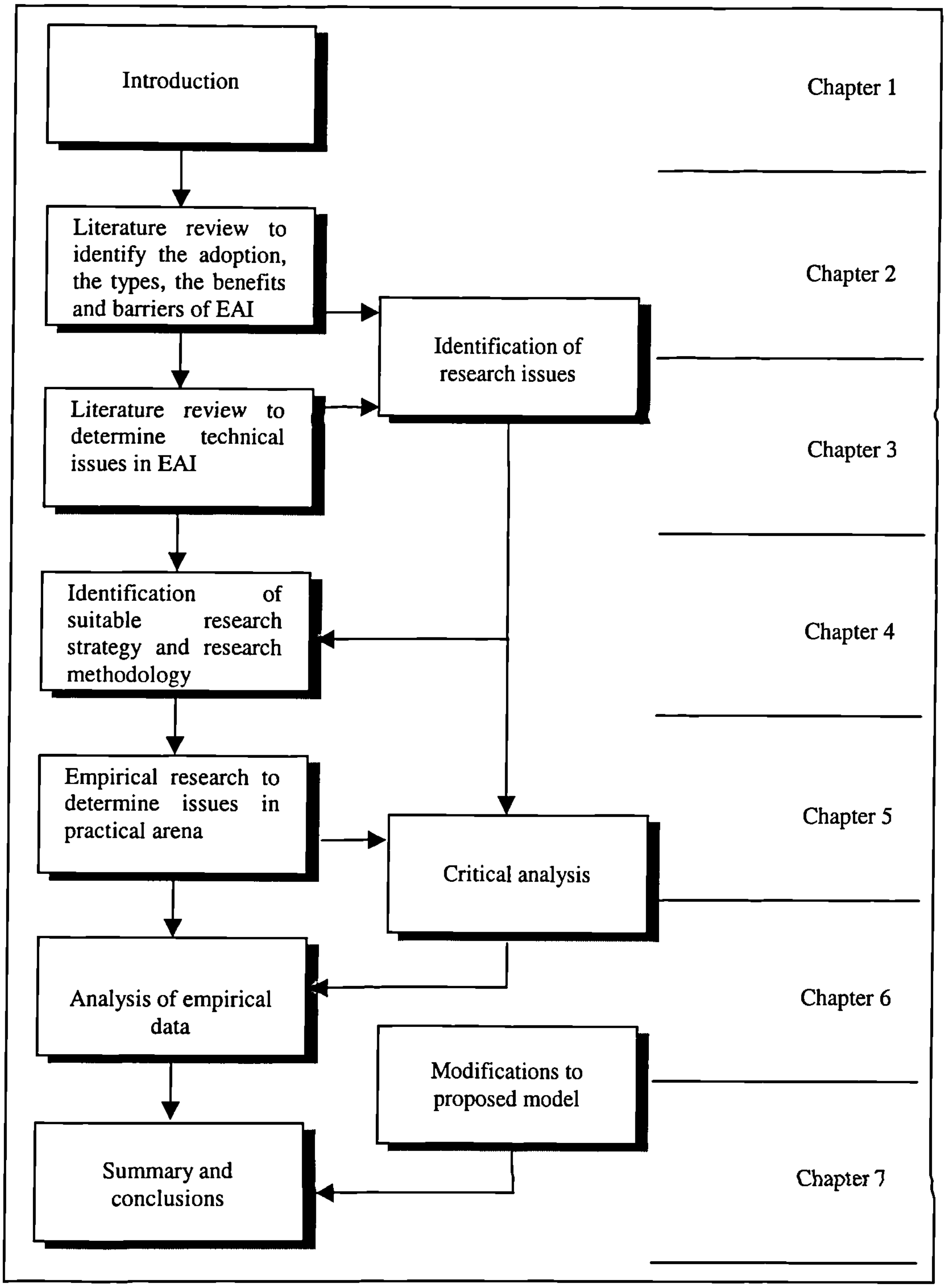

Figure 1.1: Dissertation Outline

Chapter 5: The Issues in Practice - Data Theory

Having obtained an understanding of all the relevant issues for this research, the dissertation then provides a description of the case studies studied for this research. In this context, two multinational organisations are studied and their attempts to develop a global integrated IT infrastructure are reported. Chapter 5 provides a background to the organisations and describes and analyses the main issues including: (a) the motivations to EAI adoption; (b) the adoption process; (c) the evaluation of integration technologies; (d) the pilot case studies; (e) EAI benefits, barriers and costs and (f) the global EAI solution.

Chapter 6: EAI Adoption Model – Novel Contribution

Based on the case studies and the research findings, Chapter 6 revises the conceptual model proposed in Chapter 3. The revised model supports the adoption of EAI and is influenced by ten factors including: (a) EAI costs; (b) EAI benefits; (c) EAI barriers; (d) external pressures; (e) internal motivations; (f) support; (g) the level of IT sophistication; (h) the limitations of existing IT infrastructure, (i) an evaluation framework that supports organisations to assess integration technologies and, (j) the product selection process.

Chapter 7: Summary and Conclusions – Novel Contribution

In drawing the discussion to a close, Chapter 7 summarises the research presented in this dissertation. The novel contribution is also identified in this chapter. Additionally, it provides the major conclusions reached about the possible limitations of the research and describes and discusses potential areas of further research.

2

Chapter 2: Literature Review

Summary

The aim of this chapter is to present a critical review of enterprise application integration literature. In doing so, this chapter presents: (a) a novel taxonomy for categorising types of EAI; (b) a classification of system types that are integrated; (c) a classification of EAI costs and, (d) classifications for EAI benefits and barriers. Two of these classifications map benefits and barriers against the types of systems that are integrated as well as the mode of integration adopted (e.g. process centric). Another two classifications present benefits and barriers according to the model proposed by Shang and Seddon (2000).

This chapter begins by explaining the motivations to enterprise application integration. Motivations stimulate the adoption of EAI and thus, the next section presents approaches to the adoption of integration technologies. Since, EAI is a relatively new research area, its adoption has not been widely discussed in the literature. Therefore, the author reviews and summarises adoption approaches from other relevant areas such as the adoption of Electronic Data Interchange (EDI). This will allow the author to draw parallels between EDI and EAI, and support the identification of research issues to study. The absence of theoretical models that discuss the adoption of EAI presents a theoretical gap, which is identified in section 2.2 as an issue for further investigation. This theoretical gap will be addressed in Chapter 3 by proposing a conceptual model for EAI adoption.

Section 2.3 starts introducing enterprise application integration by describing *how* information systems can be integrated through EAI. Section 2.3 focuses on: (a) understanding the types of information systems that are unified through EAI and, (b) clarifying the confusion surrounding EAI area. In doing so, the literature is reviewed and a novel taxonomy is proposed that categorises EAI, into *intra-organisational, inter-*

organisational and hybrid EAI categories. Such a taxonomy will allow readers to see through the applicability of EAI in terms of systems that are incorporated, and helps them to understand *what types of information systems EAI integrates.* In addition, this chapter explains, defines and classifies these system types that are integrated together using EAI.

Enterprise application integration benefits are then discussed and categorised. Initially, EAI benefits found in published case studies are presented and correlated with: (a) the types of systems that were integrated and, (b) mode of integration that was adopted. Application integration benefits found in the normative literature are then classified according to the model proposed by Shang and Seddon (2000). In doing so, benefits are categorised into: (a) operational; (b) managerial; (c) strategic; (d) IT infrastructure and, (e) organisational. Likewise, EAI barriers are then analysed and categorised based on Shang and Seddon's (2000) model. Along similar lines the costs that are related with the adoption of enterprise application integration are analysed and categorised based on Hochstrasser (1992) classification.

2.1 Motivations to Enterprise Application Integration Adoption

The previous chapter has discussed the limited scope and success of ERP systems and the important role of EAI in the e-business era. These issues present motivations to enterprise application integration adoption. To better understand the reasons that push organisations to turn to EAI, this section summarises the main motivations to EAI adoption.

Organisations are turning to enterprise application integration for various reasons including among others, the following:

- **Enterprise Resource Planning (ERP) systems failure** to fully automate and integrate organisations, since ERP coexist alongside other applications. ERP systems are not designed to collaborate with existing or new systems (Loos, 2000). In support of this, Davenport (1998) and Sumner (1999) report that the customisation of ERP systems is a difficult and risky task. This is attributed to that customisation problems do not allow companies to make significant changes on ERP package and thus, inhibit their integration with other existing or new applications. This is also supported by Holland and Light (1999), who mention that the $^2/_3$ of organisations that adopt ERP systems make limited changes to ERP package. As a result, organisations often change their way of doing business to fit ERP package and its philosophy. This limitation regarding ERP customisation has led organisations to bankruptcy (Sumner, 1999; Themistocleous *et al.*, 2001).

- **Technical reasons.** The non-integrated nature of IT infrastructure causes numerous problems to organisations, which need to unify their information systems and fully automate their business processes. Enterprises consist of multiple systems that in many cases replicate in functionality. This situation becomes more serious and complicated after a merger or acquisition, as in the majority of cases the IT infrastructures are incompatible. Therefore, there is a need for a technology that results into a flexible, manageable and maintainable integrated IT infrastructure (Ring and Ward-Dutton, 1999; Ruh *et al.*, 2000).

- **Financial reasons.** Organisations tend to reduce costs to improve their financial measurements. In this context, there is a need to reduce the costs of running a non-integrated IT infrastructure as well as to reduce the redundancy of data and systems. According to Edwards and Newing (2000), EAI eliminates the redundancy of data and applications and therefore, reduces operational costs since less effort is required to co-ordinate and maintain systems. Kalakota and Robinson (2001) suggest that a non-integrated infrastructure often results in a loss of sales, which also has a negative impact on the organisation. Integration supports enterprises to better co-ordinate their internal and external supply chains and reduces any loss of sales. Integration is also needed to increase enterprise productivity and performance, which results in improvements of financials measurements.

- **Managerial reasons.** The need to upgrade decision-making process and support management with real-time data implies the development of integrated IT infrastructures (Zahavi, 1999). The limitations of existing infrastructures inhibit management to take accurate decisions. The reasons for this include: (a) systems heterogeneity; (b) data redundancy and, (b) low data quality. For instance, multiple applications store data for the same entity (e.g. sales) but there is often an inability to combine data and take decisions since there is: (a) data incompatibility; (b) confusion regarding data latency, or (c) communication problems (e.g. applications can not communicate and exchange data due to their nature).

- **Strategic reasons.** Enterprises seek new ways to gain competitive advantage and believe that integration will support this strategy. For that reason, organisations are turning to EAI to fully automate their business processes and integrate their IT infrastructures (Brown, 2000). In many cases, increased competition pushes organisations to improve their productivity and easily adapt to the changing business environment. In addition, there is pressure from trading partners (customers and suppliers) that demand closer collaboration and therefore, enterprises are looking for new practices to better co-ordinate cross-enterprise business processes.

2.2 Adoption of Integration Technologies

Clearly, the motivations presented in the previous section indicate the need for the adoption of enterprise application integration. An attempt to review the literature on EAI adoption was not successful since EAI is a new research area with many gaps in normative literature. For this reason there is an absence of theoretical models and research regarding its adoption. At this point, this absence of theoretical models for EAI adoption is identified as a research issue for further investigation.

Due to EAI literature limitations, a critical review on other relevant areas that support the adoption of integration was performed. An example of such an area is Electronic Data Interchange (EDI) adoption, with models proposed by Iacovou *et al.*(1995), Van Heck and Ribbers (1999), Zinner (1999) Chwelos *et al.* (2001) and Ling (2001). This will support the author in drawing parallels among EAI and EDI. As a result, the author will be able to adapt factors from other relevant areas (e.g. EDI) to conceptualise a model for EAI adoption.

EDI is one of the first technologies that have been used to automate and integrate inter-organisational business processes. Electronic data interchange refers to the exchange of standardise messages among two applications. It is estimated by Forrester Research (1998) that by the year 2003, 80% of business-to-business integration will be achieved through EDI technology, which proves that EDI can be considered as an integration technology. EDI and EAI have similarities with both of them supporting inter-organisational integration and following the same integration concepts.

The basic EDI technical concepts focuses on: (a) extracting; (b) translating; (c) formatting and, (d) exchanging data between disparate applications using computer networks (Emmelhainz, 1993). As can be seen in Figure 2.1 *the same concepts are used by application integration technology* but in a more advanced way (Themistocleous *et al.*, 2001). Application integration extracts data or other application elements from a data-source (e.g. database, application, etc). Data extracted from applications are formatted based on a common standard and send to the network. In EDI based applications, data are extracted from a database and translated and mapped into an EDI format using an EDI standard (e.g. UN/EDIFACT). Both EAI and EDI use messages to transfer data from one source application to target. In the case of EDI, the messages are sent to the EDI manager, which distributes the right messages to the right application. In the case of application integration, application elements are sent to a central integration infrastructure that translates and reformats them into a format that is recognisable by the target application. Thereafter, the integration infrastructure routes the application elements to the target application, which simply receives them and triggers relevant tasks. In contrast to EAI, EDI messages are transmitted to target application, which translates them before proceed. This means that each EDI application should be able to understand and translate the data that are received (or send) from (to) all other EDI based applications. This requires changes to applications code, which makes EDI technology invasive[1] since each application requires changes in order to understand EDI messages. Thus, *if* an EDI application requires changes, all inter-connected applications need to be altered.

[1] Invasive is a technology that causes changes to an application's code. Non-invasive is a technology that requires no or limited changes to an application

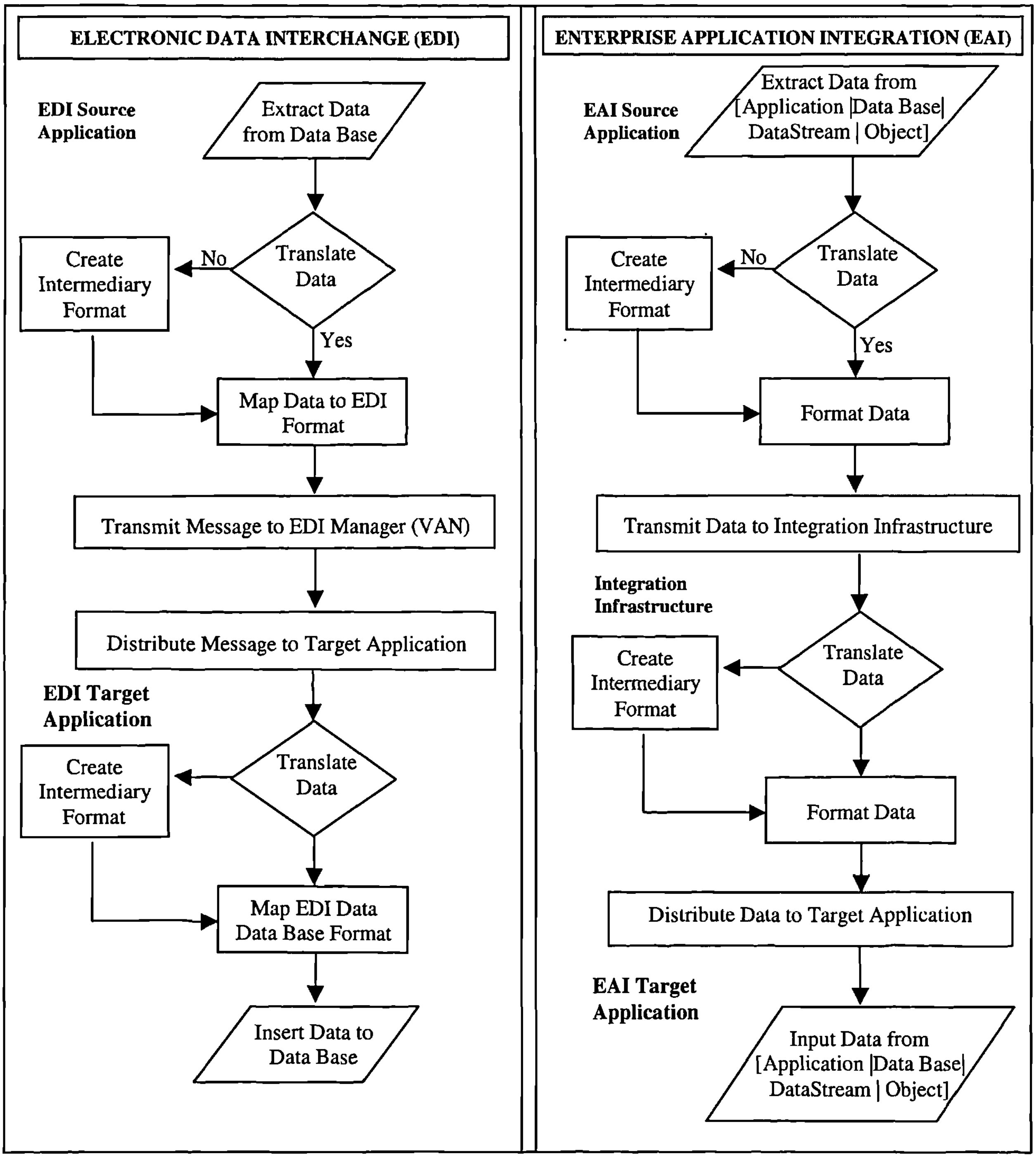

Figure 2.1: Parallels between EAI and EDI

Application integration achieves integration in a more flexible and advanced way (e.g. by using non-invasive technologies). In addition, EDI limitations like its: (a) complexity; (b) invasive nature (requires changes at both source and target application) and, (c) high cost have led integrators to steadily replace it with other more advanced and flexible integration technologies such as Extensible Markup Language (XML) (Radding, 1999). XML can therefore be considered as an integration technology with significant role in EAI.

Many models were proposed in the normative literature for the adoption of information technology such as Rogers (1995), Brancheau and Wetherbe (1990), Chung and Snyder (2000), Ling (2001) adoption models. Among others, Iacovou *et al.*(1995) identified major factors that influence the adoption of EDI technology in small and medium size (SME) enterprises. By combining and anticipating the effects of these factors they developed a model for the adoption of EDI and its integration. According to Iacovou *et al.*(1995) these major factors include:

- **Perceived benefits,** which are separated into direct (e.g. operational saving) and indirect (better customer service, improved trading partner relationships). Perceived benefits refer to the anticipated advantages that EDI technology can provide the organisation.

- **Organisational readiness for EDI** that measures whether an organisation has sufficient technological and financial resources to undertake the adoption of EDI.

- **External pressure to adopt EDI,** which refers to the influences from the environment. In the case of EDI, external pressure may include: (a) competitive pressure and, (b) imposition by trading partners.

Iacovou *et al.*(1995) also identified several factors that inhibit the adoption of EDI, such as: (a) the complexity of the technology; (b) the need to change internal systems; (c) the lack of technological skills and, (d) the lack of system integration. Although they expect these factors to play a significant role in the adoption of EDI, Iacovou *et al.*(1995) did not include them into their model. Their explanation for this is that these factors are generally identified through studies of large organisations and therefore, their applicability in SME's is questionable.

Ling (2001) consider the factors reported by Iacovou *et al.*(1995) for inhibiting the adoption of EDI as barriers that restrict the adoption of electronic commerce. Zinner (1999) studied the barriers to EDI and suggests that the lack of consensus among existing EDI standards is an obstacle to EDI adoption. The reason for this is that many standards exist (e.g. ANSI X12, UN/EDIFACT, proprietary standards) that make it hard for an SME to implement EDI. In this context, Van Heck and Ribbers (1999), propose that the availability of EDI standards is a factor that affects the adoption of EDI. As a result, Van Heck and Ribbers (1999) expand the model proposed by Iacovou *et al.* (1995) through incorporating this factor (availability of standards).

Many other factors that influence the adoption of IT are proposed in the literature, with Sumner and Holstetler (1999) suggesting among others that support (e.g. vendor support, manager support) and environmental (competition, pressure) factors are important for the adoption of innovation.

Chwelos *et al.* (2001) further analysed the factors that influence the adoption of EDI and revised Iacovou *et al.*(1995) model by focusing on inter-organisational systems. According to Chwelos *et al.* (2001), perceived benefits, organisational readiness and external pressure deal with technological, organisational and inter-organisational perspectives respectively. In addition, Chwelos *et al.* (2001) support that no single study has tested a model that comprehensively addresses the technological, organisational and inter-organisational levels.

Chwelos *et al.* (2001) augmented Iacovou *et al.*(1995) model by incorporating new and existing factors from the literature that influence external pressure and organisational readiness. These factors include: (a) competitive pressure; (b) dependency on trading partner; (c) enacted trading partner power and, (d) industry pressure influence external pressure. Likewise, (e) financial resources; (f) IT sophistication and, (g) trading partnership readiness affect organisational readiness. This indicates that the factors reported by Iacovou *et al.*(1995) are not the only one that affect EDI adoption. Other factors like those reported in the aforementioned studies influence EDI adoption. As a result, Figure 2.2 presents Iacovou *et al.*(1995) model incorporated with Chwelos *et al.* (2001) and Van Heck and Ribbers (1999) augmentations. In doing so, the author suggest that all these factors may also influence EAI adoption.

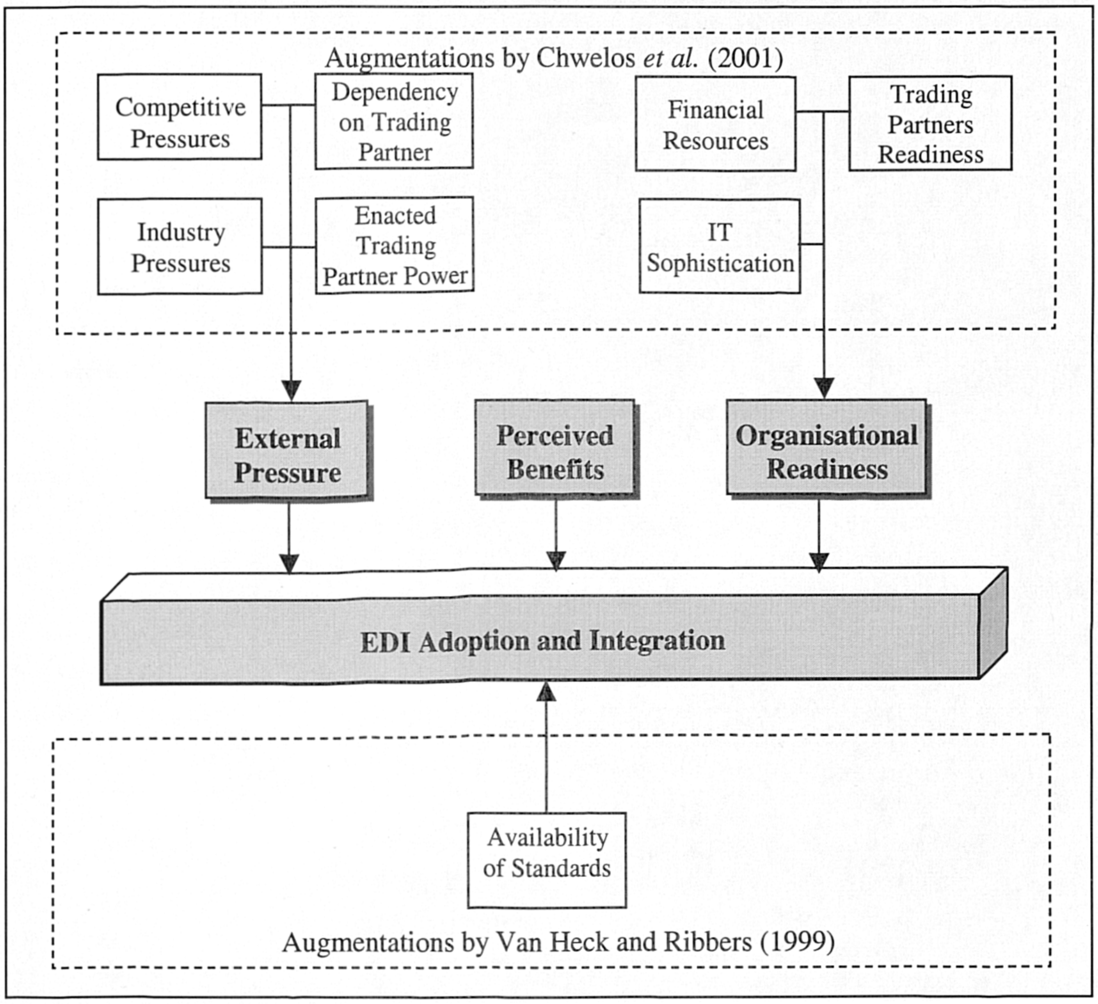

Figure 2.2: Iacovou Model for EDI Adoption and Integration

2.3. A Novel Taxonomy for Classifying Types of Application Integration

Having discussed the need for integrating IT infrastructures and reported relevant background theory on integration adoption, the present and next sections introduce the main research area of this dissertation, which is enterprise application integration. This section analyses the types of IS that unified through application integration. In addition, different types of EAI are classified

Confusing terminology in the integration area has led to a debate regarding the types of information systems that can be integrated through EAI, as each term proposes the incorporation of different types of systems (e.g. ERP-to-legacy systems integration). For example, Grimson *et al.*(2000) suggests that the term Enterprise Application Integration (EAI) refers to the integration of Enterprise Resource Planning (ERP) systems (e.g. ERP to ERP), while Duke *et al.*(1999) claim it supports the incorporation of all packaged applications. In contrast, Ruh *et al.*(2000) report EAI does not only piece together packaged systems but also intra-organisational solutions, while Zahavi (1999) claims that it supports both enterprise and cross-enterprise application incorporation. Themistocleous *et al* (2000) have attempted to overcome much of application integration terminology confusion by evaluating the various definitions and proposing common terminology and approaches for EAI.

Clearly, there is a need to better clarify this confusing terminology and define the dimensions-types of application integration (i.e. intra-organisational EAI). In addressing the aforementioned need, *a novel taxonomy* is proposed by the author, which will clarify this confusion. The taxonomy is based on the critical analysis and evaluation of existing case studies and associated literature on EAI. The *novelty of the taxonomy focuses on the synthesis of a comprehensive set of systems that efficiently describe the dimensions of EAI applications.* The proposed taxonomy will allow managers and developers to better understand the integration area, and can be used as a tool for decision-making. Based on this novel taxonomy, managers and business analysts will be able to interpret and apprehend the capabilities of EAI technology. As a result, understanding that EAI unifies both enterprise and cross enterprise applications and therefore, leading to the development of an integrated infrastructure that supports intra-organisational and inter-organisational applications (Brown, 2000; Kalakota, 2000). Thus, the proposed taxonomy helps in decision making when organisations adopting integration technologies or designing integrated infrastructures.

Normative literature classifies information systems into intra-organisational and inter-organisational. Bytheway and Dhillon (1996) and Kaufman (1996) report that inter-organisational systems (IOS) are networks of systems that allow businesses to share information and interact electronically across organisational boundaries. In contrast, intra-organisational solutions exchange data at an enterprise level (Emmelhainz, 1993). It is therefore suggested by the author, that EAI might follow this classification. In support of this, Zahavi (1999) suggests that EAI incorporates enterprise and cross enterprise applications, and therefore leads to integrated intra-organisational and inter-organisational systems. For instance, the integration of the systems that co-ordinate an internal supply chain presents an intra-organisational integrated system. Similarly, the integration of an external supply chain reflects inter-organisational applications integration. Hence, the proposed taxonomy separates EAI into *intra-organisational* and *inter-organisational EAI*. In addition, a new sub-category called *Hybrid EAI* is also proposed and integrated into the proposed taxonomy, and is used to describe e-commerce applications that function as intra-organisational and/or inter-organisational EAI systems. Figure 2.3 demonstrates the proposed novel taxonomy.

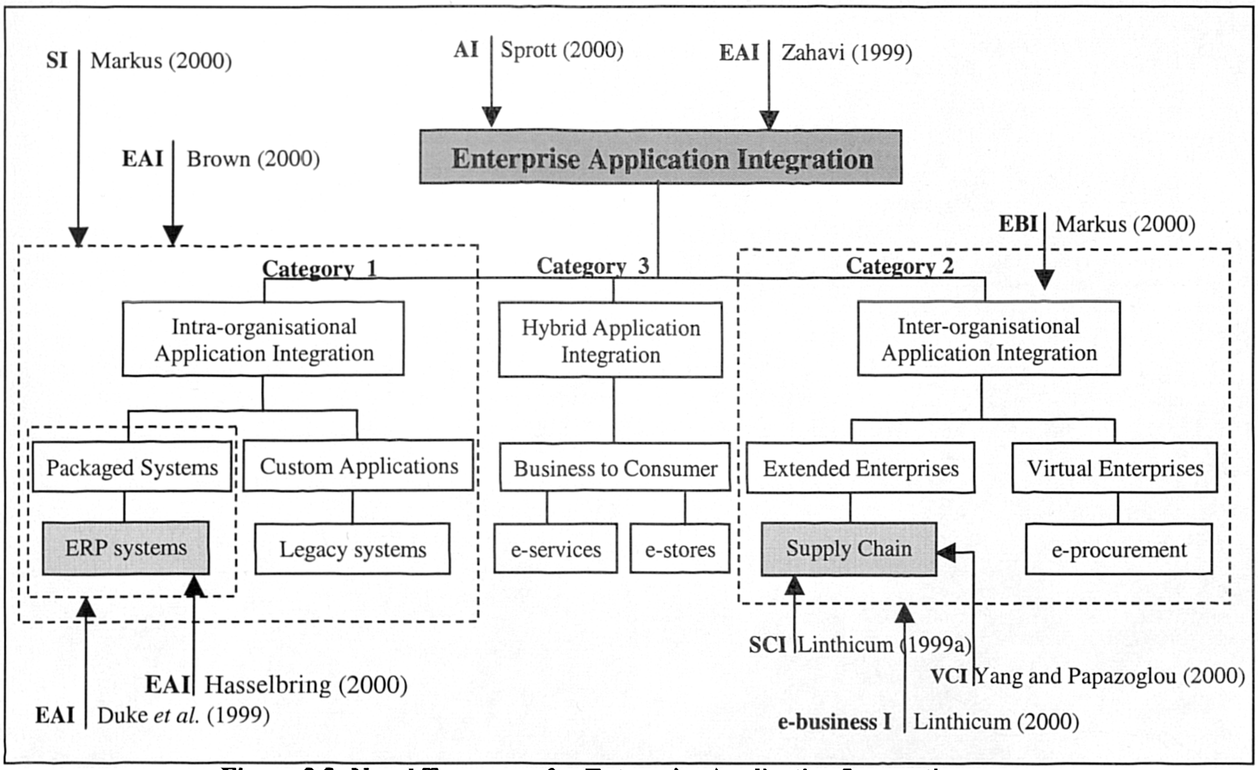

Figure 2.3: Novel Taxonomy for Enterprise Application Integration

The taxonomy classifies EAI into: (a) intra-organisational; (b) inter-organisational and, (c) hybrid EAI. At a second level, intra-organisational EAI is sub-divided into packaged systems and custom applications integration where hybrid EAI, into Business to Consumer (B2C) integration. Inter-organisational EAI is further categorised into extended and virtual enterprises integration. The lowest level in Figure 2.3 presents exemplar applications for each category (e.g. e-stores, e-procurement). One of the novelties of Figure 2.3 is that it correlates the confusing terminology on integration within the proposed taxonomy. The following sub-sections discuss the proposed taxonomy.

2.3.1 Category 1: Intra-organisational EAI

Packaged and custom systems are classified as intra-organisational applications and thus, they form subcategories of intra-organisational EAI (Handfield and Nichols, 1999). Brodie and Stonebraker (1995) report that custom applications such as legacy systems were developed to operate in a particular way and thus, resist modification and evolution to meet business requirements. Therefore, a custom system designed for one company can not be adopted by another company. Most legacy systems follow a monolithic model (Zahavi, 1999) in which data, logic and interfaces are not separated but are built together (Bernus *et al.*, 1996). In contrast to custom systems, packaged solutions follow a three-tier architecture model in which data are separated from business logic and interfaces and they can therefore easily updated or modified (Wijegunarate and Fernandez, 1998; Serain, 1999).

In addition, packaged systems like Enterprise Resource Planning (ERP) solutions were based on generic business requirements and processes, and not on the requirements of a specific organisation (Loinsky, 1995; Holland and Light, 1999; Holland *et al.*, 1999). Thus, the same packaged system (e.g. SAP) is adopted by thousands of enterprises around the world, without much customisation. However, Davenport (1998) reports that packaged systems do not allow much customisation, and thus, organisations have to change their business processes and strategy to fit packaged systems. This may have an impact on competitive advantage of a company since multiple organisations follow similar strategies.

Case studies reported by Edwards and Newing (2000) demonstrate that application integration efficiently supports the integration of ERP systems and custom applications using a diversity of integration technologies such as database oriented middleware[2] (e.g. ODBC), messaging technologies (e.g. XML, message brokers), transaction oriented technologies (e.g. application servers), distributed objects (e.g. CORBA) and interface oriented technologies (e.g. adapters, wrappers). However, both packaged and custom systems have different types of integration problems (e.g. in retrieving data) and therefore, focus on different integration technologies[3] (Themistocleous and Irani, 2002b). As a result, packaged and custom systems can form two different subcategories of intra-organisational EAI

2.3.2 Category 2: Inter-organisational EAI

Inter-organisational integration seeks to incorporate cross-enterprise business processes and systems such as supply chains (Brown, 2000). E-business solutions are part of this subcategory with Kalakota and Robinson (1999) classifying them as inter-organisational applications. Linthicum (2000) suggests that EAI incorporates e-business through the same category of technologies (e.g. message brokers, adapters, XML) which support intra-organisational integration. Much literature (Loinsky, 1995; Brown, 2000; Puschmann and Alt, 2001) classifies integrated applications according to the degree (loose or tight) of integration achieved. This categorisation is important, as companies tend to follow the loose or the tight degree of integration when incorporating their e-business systems (Ring and Ward-Dutton, 1999).

According to Helm (1999), loose integration is often followed by loosely coupled trading partners. These partnerships select loose integration to simply share or exchange information electronically. Puschmann and Alt (2001) report that loosely integration is correlated with asynchronous communication (see Appendix C). The type of communication determines a kind of dependencies among two applications and influences the processing sequence of the involved applications (Serain, 1999). In asynchronous communication, applications communicate over time without having to wait for target application to receive and process the data, and reply to the source application (Morgenthal and La Forge, 2000). In general, loose integration is adopted by organisations that are reluctant to tightly integrate their

[2] The integration technologies reported in this chapter are analysed in Appendix B
[3] For instance, screen wrappers are used to extract data and support custom systems integration where packaged solutions focus on Application Programming Interfaces (APIs) to achieve integration.

systems over cross-enterprise networks due to security, cultural and control reasons. Alternatively where there is no need to develop a tight integration model, which results in the development of a common inter-organisational integration infrastructure (Helm, 1999).

Tightly integrated applications are characterised by a higher degree of process dependency. According to Puschmann and Alt (2001) and Ruh *et al.*(2000), tightly integrated applications follow synchronous communication (see Brodie and Stonebraker, 1995; Wijegunarate and Fernandez, 1998) with the sender application pausing its operations and waiting for the receiver to execute senders request or, process the data requested and reply. This type of communication is accomplished in a co-ordinate manner, which may lead integrated applications to fail if one system is unable to execute a process. In this case, all partners fail to complete their processes. The reason for this is that all participate in the same logical business process (Linthicum, 1999b; Kalakota, 2000). Helm (1999) suggests that organisations develop a homogeneous inter-organisational IT infrastructure when selecting tight integration since such infrastructures allow them to increase their efficiency and function as a 'single' (virtual) enterprise. Table 2.1 illustrates the characteristics of both loose and tight types of integration.

Loose Integration	**Reference**
Focuses on exchanging-sharing data among partners	Kalakota and Robinson (1999)
Low degree of processes dependency	Loinsky (1995)
Low degree of integration	Brown (2000)
The development of a homogeneous integrated cross-enterprise infrastructure is **not** important	Helm (1999)
Asynchronous communication	Puschmann and Alt (2001)
Tight Integration	**Reference**
Focuses on integrating cross enterprise business processes and systems	Themistocleous and Irani (2001a)
Highest degree of processes dependency	Kalakota and Robinson (1999)
High degree of integration	Brown (2000)
The development of a homogeneous integrated cross-enterprise infrastructure is important	Helm (1999)
Synchronous communication	Puschmann and Alt (2001)

Table 2.1: Loose and Tight Integration

Based on the degree (loose, tight) of integration, Helm (1999) proposed three scenarios for e-business integration, which include:

- enabling extended enterprises;
- enabling virtual enterprises and,
- e-commerce EAI.

The first scenario, *enabling extended enterprises,* refers to loosely integrated e-business applications (e.g. e-Supply Chain Management) where the need for the development of a homogeneous cross-enterprise integrated infrastructure is not important. Organisations simply extend their business activities through e-business solutions and try to loosely incorporate these applications with external partners (Riggins and Rhee, 1998; Helm, 1999). In this case, collaborators exchange data without sharing a common IT infrastructure or business processes. For instance, a retailer sends an electronic order to its supplier. The supplier, checks its stocks and if there is availability it fulfils the order and sends the invoice to the retailer. In such an instance, suppliers do not have access to their retailer's IT infrastructure, to check the orders or monitor the whole process.

The scenario *enabling virtual enterprises* refers to tightly integrated e-business applications where integration is very important with a number of enterprises sharing common data and processes. In doing so, attempting to function as one (virtual) organisation. In many cases to support more efficient the common processes, real-time information is needed. As a result, a high degree of incorporation is required between back-end systems and e-business solutions to support real-time information. For instance, a food retailer and its suppliers may integrate their IT infrastructures to control and improve promotion management. Suppliers might gain access to the IT infrastructure of the retailer and retrieve information relating to their own products and promotions. Suppliers could analyse the availability and sales of their products and, replace them according to the agreement they have with the retailer. In such a scenario, both suppliers and retailer share common business processes and IT infrastructures.

The first two scenarios proposed by Helm (1999) (enabling extended enterprises, enabling virtual enterprises) are adopted to further classify inter-organisational EAI as they both deal with the integration of e-business solutions. However, the differences that exist between these two scenarios and presented in Table 2.1 have led the author to separate them in two different categories (extended enterprises and virtual enterprises).

2.3.3 Category 3: Hybrid EAI

The third scenario (e-commerce EAI) proposed by Helm (1999), presents no challenge for integration among business partners. The reason for this is that it focuses on Business-to-Consumer (B2C) solutions and therefore, no business partners involved in these types of systems. However, literature (Bakos, 1998; Lee, 1998; Lohse and Spiller, 1998; Riggins and Rhee, 1998; Timmers, 1999; Anonymous, 2001) supports that in some cases (e.g. e-stores) *there is a need* to integrate B2C applications with other inter-organisational solutions (e.g. suppliers, distributors, bank etc). The reason for this is that such inter-organisational systems have an important role in supporting the functionality of an e-commerce application. As a result, part of business-to-consumer applications function as inter-organisational systems while others as intra-organisational applications.

The main users of B2C applications include a company that owns the application (service provider, shop-provider) and internet-users (consumers) that communicate with B2C solutions (Doukidis *et al.*, 1998). In some applications (e.g. e-services) consumers subscribe once (by paying a fixed amount of money to a bank) and then use the system for a specific time period (e.g. one year). During this period, the owner of B2C applications provide services to the customer without the involvement of an external entity (e.g. supplier). Thus, there is no need to integrate this type of systems with external partners-companies, as there are no external companies (trading partners). This type of system functions like an intra-organisational application. However, other types of B2C applications function like extended enterprises or virtual enterprises. For example, e-store applications require integration across enterprises, as the e-store incorporates banks', suppliers' and distributors' systems. Hence, the proposed taxonomy adopts a new sub-category (Hybrid EAI) at the same level as intra and inter-organisational and includes B2C EAI. Table 2.2 summarises the characteristics of each category of the proposed taxonomy.

Category	Characteristics	References
Intra-organisational EAI	Integrates enterprise applications	Brown (2000) Loinsky (1995)
	Integrates packaged and custom systems	Edwards and Newing (2000) Ruh *et al.* (2000)
	No transactions with external users or partners	Themistocleous and Irani (2000) Helm (1999)
Hybrid EAI	Integrates business to consumer applications with IT infrastructure	Themistocleous and Irani (2000)
	Internet users purchase products or services. Hybrid EAI applications support the transactions by integrating internal systems or/and external partners	Kalakota and Robinson (1999) Doukidis *et al* (1998))
Inter-organisational EAI	Integrates cross-enterprise applications with IT infrastructure	Linthicum (2000) Zahavi (1999)
	Integrates business-to-business applications	Markus (2000) Morgenthal and La Forge (2000)
	Based on the degree (loose, tight) of integration it is separated: ▪ Extended enterprises (loose integration) ▪ Virtual enterprises (tight integration)	Helm (1999)

Table 2.2: Characteristics of the Sub-categories of the Proposed Taxonomy

2.4 A Classification of System Types that are Integrated

Kalakota and Robinson (2001), Kalakota (2000) and Wijegunarate and Fernandez (1998) among others indicate that companies integrate disparate systems when adopting EAI solutions. Such systems have been discussed by Chung and Snyder (2000) and Brodie and Stonebraker (1995), and are classified into custom applications, packaged systems and e-business solutions. Puschmann and Alt (2001), Edwards and Newing (2000) and Zahavi (1999) report that from a technical perspective, organisations combine these three types of systems in various ways when being integrated. *However, these permutations have not yet been described and identified in the application integration literature.* The author has conducted an extensive review of the normative literature and analysed published case studies in the application integration area. In doing so, identifying permutations of system types that are pieced together when EAI is used. Table 2.3 illustrates that organisations integrate the three aforementioned types of systems (custom, packaged and e-business solutions) by making all unique permutations. Table 2.3 maps the system types that were integrated in each case study. Based on these permutations, the author defines classifications of system types that are integrated through EAI. Table 2.4 describes these classifications.

		Classifications of System Types						
		Custom to Custom	Custom to Packaged	Custom to e-business	Packaged to Packaged	Packaged to ebusiness	Ebusiness to ebusiness	Custom to Packaged to ebusiness
Case Studies	**Reference**							
Deutsche Bank	Edwards and Newing (2000)	✓		✓			✓	
Amazon. Com	Anonymous (2001)					✓	✓	
British Airways	Edwards and Newing (2000)	✓						
EDS Enterprise Solutions	Edwards and Newing (2000)		✓		✓			
National Power – London	Anonymous (1999b)	✓						
Elsevier Science	Edwards and Newing (2000)			✓				
VF Corporation	Edwards and Newing (2000)				✓	✓		✓
XEROX LTD	Selsikas (1999)		✓					✓
Fujitsu Computers	Edwards and Newing (2000)		✓					
PETs Mart	Anonymous (1999c)		✓	✓				
General Motors	Edwards and Newing (2000)	✓		✓	✓		✓	
Scottish Power	Edwards and Newing (2000)		✓		✓			✓
Catawba Memorial Hospital	Anonymous (1999a)	✓						
Honeywell Europe	Edwards and Newing (2000)	✓	✓		✓			
Tesco	Edwards and Newing (2000)			✓				
Bosch Group	Puschmann and Alt (2001)	✓			✓			✓
Zurich Financial Services	Edwards and Newing (2000)	✓						

Table 2.3: Classifications of System Types that are Integrated through EAI

Classifications of System Types	Description
Custom to Custom Integration	Custom applications like legacy applications and data warehouses are frequently integrated in a common infrastructure, to fully automate business processes. A typical scenario of this classification could be the incorporation of legacy systems that deal with promotions management (e.g. stocks, suppliers accounts).
Custom to Packaged Integration	This is a common approach when organisations adopt EAI since packaged applications like Enterprise Resource Planning (ERP) systems have in many cases failed to achieve integration and co-exist alongside custom applications. A typical scenario of this type could be the integration of a legacy system that deals with production and an ERP module that handles customer orders or suppliers' details/accounts.
Custom to e-business Integration	Many e-business solutions require a close collaboration with legacy applications to support e-business enabled processes and tasks. As a result, custom applications (e.g. stocks) are incorporated with e-business systems to integrate and automate inter-organisational business processes. In many cases the functionality of an e-business solution is used to support custom systems. For instance, an e-store updates a custom system that deals with stock availability. The information provided by the e-business solution is critical not only for the functionality of stock application but, also for the whole supply chain as it supports the automation and integration of specific business processes.
Packaged to Packaged Integration	In this case, disparate packaged systems such as different versions of an ERP system or many ERP modules that exist in one organisation are unified into a common integrated infrastructure. For instance, after a merger or acquisition there is a need to integrate the various ERP systems that exist both in mother company and its subsidiaries.
Packaged to e-business Integration	Organisations take advantage of EAI and Electronic Commerce technology when they integrate their e-business solutions with packaged applications as ERP systems can be used as back-office system to support the e-business functionality (front end application). In this case, processes that usually deal with e-sales, e-procurement and e-supply chain management can be integrated with packaged systems.
E-business to e-business Integration	In this approach, an e-business application is integrated and supports the functionality of another e-business solution. For example an electronic point of sales is incorporated with e-supply chain management to share data that are important for the latter application (e.g. customer orders, customer details etc).
Custom to Packaged to e-business Integration	Such approaches focuses on the development of an integrated infrastructure that integrates processes and applications on departmental, enterprise or cross-enterprise level. For instance, an estore is integrated with the financial module of an ERP system and a legacy system that deals with stocks availability.

Table 2.4: Classifications of System Types that are Integrated

2.5. Benefits of Application Integration

Published case studies on application integration area (Edwards and Newing, 2000; Puschmann and Alt, 2001) suggest that the adoption of EAI solutions provide significant benefits to organisations. From a technical perspective, EAI achieves data, objects, interfaces and processes integration. The reason for this is that EAI overcomes integration problems at all integration levels (e.g. data level, process level etc) using a diversity of technologies such as message brokers and XML. Since the integration technologies being used eliminate changes to existing applications, they result to more flexible, manageable and maintainable solutions. These technologies are based on common standards and result in the development of a unified IT infrastructure that supports custom systems, packaged applications and e-business systems integration. Thus, EAI enables information sharing and achieves common business processes.

From a business perspective, application integration reduces the overall integration cost (Ring and Ward-Dutton, 1999). The reason for this is based on the decrement of both integration time and maintenance costs. Application integration leads to return on investment (ROI) as it provides a flexible, manageable and maintainable enterprise infrastructure that supports the changing business and technical requirements. Integrated enterprise architectures allow companies to increase their productivity and provide better services for their customers (Ruh *et al.*, 2000). Moreover, organisations improve their relationships with their clients as well as improve their performance (Urlocker, 2000). Kalakota and Robinson (2001) suggests that application integration supports strengthened supply chains and improved relationships and collaboration between organisations and suppliers. Other benefits reported by Morgenthal and La Forge (2000) include the provision of a centralised point of control, the reduction of skills required to integrate applications, faster time to marketing and increased market share.

The author has analysed 15 published case studies on integration area (Edwards and Newing, 2000; Puschmann and Alt, 2001). Based on the results of the analysis, the author summarises EAI benefits by presenting them in Table 2.5. In addition Table 2.5 maps the benefits against: (a) the types of systems that are integrated (custom, packaged and e-business) and,

(b) the mode of integration such as data centric or process centric[4] (Carrier, 1999). This classification, allow researchers to compare and better analyse the benefits when organisations follow an integration approach. For example, as illustrated in Table 2.5, enterprises that select an EAI solution to integrate their internal applications (custom and packaged) have achieved a flexible, manageable, maintainable solution, which results on ROI and reduces the cost. As the number of cases increases in the proposed classification it will provide more harmonised results, and will allow better analysis and decision-making.

[4] *Process centric* integration deals with automating, unifying and improving business processes.
Data centric integration focuses on integrating the data flows that are exchanged among applications.

	CASE STUDIES														
	British EAIrways	Catawba Memorial Hospital	National Power - London	Zurich Financial Services	PETsMart	EDS Enterprise Solutions	Fujitsu Computers	Honeywell Europe	Scottish Power	Elsevier Science	Tesco	VF Corporation	Bosch Group	Deutsche Bank	General Motors
Systems Integrated through Enterprise Application Integration															
Custom Systems	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Packaged Applications					✓	✓	✓	✓	✓			✓	✓	✓	✓
E-business and Electronic Commerce Solutions										✓	✓	✓	✓	✓	✓
Data Vs Process Centric															
Data integration	✓	✓				✓			✓					✓	
Process integration			✓	✓	✓		✓	✓		✓	✓	✓	✓		✓
Enterprise Application Integration Benefits															
Provides more understanding and control of processes	✓						✓								
Improves management and supports decision making	✓										✓				
Results in more organised business processes							✓								
Allow organisations to do business more effectively	✓										✓			✓	
Improves planning in supply chain management											✓				
Increases collaboration among partners											✓				
Reduces lost sales	✓										✓				
Increases productivity		✓													
Increases performance														✓	
Achieves customer satisfaction		✓					✓				✓			✓	
Results in reusable systems, components and data								✓		✓		✓		✓	✓
Reduces redundancy (applications, data, tasks)	✓														
Reduces cost		✓	✓		✓		✓		✓		✓		✓	✓	
Achieves return on investment		✓	✓		✓	✓			✓						
Faster and cheaper implementation than bespoke solutions						✓			✓				✓		
Increases flexibility				✓		✓	✓	✓	✓			✓			
Quicker response to change						✓									
Offers interfaces-standardisation						✓									
Provides flexible, maintainable and manageable solutions			✓	✓	✓			✓	✓	✓		✓	✓		✓
Results in reliable data										✓					
Process and systems scalability									✓						✓
Provides portability									✓						
Reduces development risks									✓						
Achieves non-invasive solutions				✓								✓			
Achieves process integration				✓											
Increases data analysis	✓		✓											✓	
Improves data quality					✓					✓					
Supports efficient data sharing			✓												

Table 2.5: Benefits of Enterprise Application Integration

2.5.1 Classification of EAI Benefits

A number of different models such as those proposed by Irani (1998) and Ward and Griffiths (1997) exist in the literature to classify the benefits of information systems. Shang and Seddon (2000) propose a model to classify the benefits that derived from integrated IT infrastructures and systems such as ERP's. This model can be adopted for the classification of EAI benefits, since EAI technology integrates IT infrastructures and automates business processes. Shang and Seddon (2000) propose the following classification as presented in Table 2.6.

Dimension	Sub-Dimension
Operational	▪ Cost reduction ▪ Cycle time reduction ▪ Productivity improvement ▪ Quality improvement ▪ Customer services improvement
Managerial	▪ Better resource management, ▪ Improved decision making and planning ▪ Performance improvement
Strategic	▪ Support business growth ▪ Support business alliance ▪ Build business innovations ▪ Build cost leadership ▪ Generate product differentiation (including customization) ▪ Build external linkages (customers and suppliers)
IT Infrastructure	▪ Build business flexibility for current and future changes ▪ IT costs reduction ▪ Increased IT infrastructure capability
Organisational	▪ Support organisational changes ▪ Facilitate Business learning ▪ Empowerment ▪ Built common visions

Table 2.6: Classification of ERP Benefits – Source Shang and Seddon (2000)

The benefits of application integration found in the normative literature are classified and summarised on Table 2.7 according to the model proposed by Shang and Seddon (2000).

Dimension	Benefits	Reference
Operational	▪ Reduces lost sales ▪ Increases productivity ▪ Achieves customer satisfaction ▪ Reduces cost ▪ Improves data quality	▪ Edwards and Newing (2000) ▪ Duke *et al.*(1999) ▪ Kalakota and Robinson (1999) ▪ Linthicum (1999a) ▪ Ring and Ward-Dutton (1999)
Managerial	▪ Provides more understanding and control of processes ▪ Improves management and supports decision making ▪ Improves planning in supply chain management ▪ Increases performance ▪ Achieves return on investment ▪ Results in reliable data ▪ Increases data analysis ▪ Provides a centralised point of control	▪ Duke *et al.*(1999) ▪ Edwards and Newing (2000) ▪ Kalakota (2000) ▪ Linthicum (1999b) ▪ Edwards and Newing (2000) ▪ Zahavi (1999) ▪ Klasell and Dudgeon (1998) ▪ Brown (2000)
Strategic	▪ Improves planning in supply chain management ▪ Allow organisations to do business more effectively ▪ Increases collaboration among partners ▪ Increased market share ▪ Improves relationships with suppliers	▪ Linthicum (2000) ▪ Brown (2000) ▪ Edwards and Newing (2000) ▪ Urlocker (2000) ▪ Ruh *et al.*(2000)
IT Infrastructure	▪ Results in reusable systems, components and data ▪ Reduces redundancy of applications, data and tasks ▪ Faster and cheaper implementation than bespoke solutions ▪ Offers interfaces-standardisation ▪ Provides flexible, maintainable and manageable solutions ▪ Results in reliable data ▪ Process and systems scalability ▪ Provides portability ▪ Reduces development risks ▪ Achieves non-invasive solutions ▪ Achieves process integration ▪ Improves data quality ▪ Supports efficient data sharing ▪ Provides data integration ▪ Provides objects/components integration ▪ Provides real-time integration ▪ Integrates custom systems ▪ Integrates packaged systems ▪ Integrates e-business solutions	▪ Zahavi (1999) ▪ Klasell and Dudgeon (1998) ▪ Edwards and Newing (2000) ▪ Morgenthal and La Forge (2000) ▪ Linthicum (2000) ▪ Zahavi (1999) ▪ Ruh *et al* (2000) ▪ Ring and Ward-Dutton (1999) ▪ Ring and Ward-Dutton (1999) ▪ Linthicum (1999a) ▪ Zahavi (1999) ▪ Zahavi (1999) ▪ Linthicum (2000) ▪ Edwards and Newing (2000) ▪ Klasell and Dudgeon (1998) ▪ Ring and Ward-Dutton (1999) ▪ Themistocleous *et al.*(2000) ▪ Morgenthal and La Forge (2000) ▪ Linthicum (2000) ▪ Kalakota and Robinson (2001)
Organisational	▪ Results in more organised business processes ▪ Allow organisations to do business more effectively ▪ Increases flexibility ▪ Quicker response to change ▪ Achieves process integration	▪ Brown (2000) ▪ Linthicum (2000) ▪ Ring and Ward-Dutton (1999) ▪ Kalakota and Robinson (1999) ▪ Linthicum (1999a)

Table 2.7 Classifications of Enterprise Application Integration Benefits

2.6. Barriers to Enterprise Application Integration

Similarly to the introduction of other technologies (e.g. ERP systems), the adoption of application integration has both benefits and barriers (e.g. employees resistance to change). This section analyses and summarises the barriers of enterprise application integration.

Despite EAI vendors promoting their products as 'plug and play' (Linthicum, 1999a), there are no 'off-the-self' application integration solutions that offer 'out-of-the-box' (automated) integration (Zahavi, 1999). From a technical perspective, EAI is based on a diversity of technologies (e.g. adapters, XML) to incorporate systems. These technologies achieve integration at different levels (e.g. data, message, object, interface and/or process level). However, there is no single integration technology efficiently supporting all integration levels (Duke *et al.*, 1999; Ring and Ward-Dutton, 1999). Some integration technologies are more effective at one level of integration, whereas others are at another level of integration. Therefore, permutations of EAI technologies are needed to overcome integration problems (Duke *et al.*, 1999).

There is much confusion regarding the permutations of integration technologies that can be used to piece together information systems. The reason for this is that there are integration technologies that overlap in functionality but differ in the quality (e.g. portability, flexibility, scalability) and efficiency of their solutions. Moreover, the majority of applications that are pieced together differs in integration requirements, which means that the permutations of integration technologies are not only based on their functionality, but also on integration requirements and constrains. Application integration solutions are based on a combination of technologies, as no single EAI product addresses all integration problems (Ring and Ward-Dutton, 1999).

Edwards and Newing (2000) suggest that the product choice and maturity of integration technologies also forms a barrier to EAI. Since, integrators combine a variety of technologies and products to achieve integration, a knowledge regarding the capabilities of each technology and product is required. However, there is a lack of knowledge about the 'real' capability of each technology thus, making the product selection more difficult (Ruh *et* *al.*, 2000). Some integration technologies remain immature which also becomes a barrier to application integration. Moreover, some integration technologies are not appropriate to integrate specific types of systems. For instance screen wrappers efficiently support the integration of custom systems but are not appropriate for e-business or packaged applications integration (Zahavi, 1999).

According to Markus (2000) and Ring and Ward-Dutton (1999), EAI requires a vast amount of technical expertise and a complex set of skills. However, there is a lack of skilled staff who are familiar with application integration. Ruh *et al.*(2000) report that EAI requires different skills than those that have traditionally been available in IT groups. Apart from knowledge of integration techniques and tools, IT staff should have also knowledge of middleware technologies such as database oriented, message oriented, transaction oriented and distributed objects oriented middleware (Ring and Ward-Dutton, 1999; Ruh *et al.*, 2000). Moreover, the shortage of skilled staff and the high-skills required have led to high salaries for integrators, which translates in higher project cost (Zahavi, 1999). The cost of integration could also be a factor to EAI as many companies have considered integration as a major investment (Edwards and Newing, 2000).

The organisational culture, is often a barrier to application integration since previous data integration failures have created negative perceptions to be overcome (Brown, 2000). From another perspective, organisational culture also forms a barrier to EAI since some enterprises or their departments are reluctant to share their business data or processes with other departments within the same company or with external partners (Brown, 2000). In some cases enterprises claim that are reluctant to share their data for security issues (do not allow the sharing of business data over open networks such as Internet). In other cases, enterprises or departments feel reticent to share their data since they believe that they will become weaker. For these departments or organisations the ownership and the control of business data and processes is related with their power. As a result, politics also form a barrier to application integration.

Table 2.8 correlates the barriers to EAI that derive from the analysis of 15 published EAI case studies with the types of systems that were integrated and the mode of integration.

	CASE STUDIES														
	British EAIrways	Catawba Memorial Hospital	National Power - London	Zurich Financial Services	PETsMart	EDS Enterprise Solutions	Fujitsu Computers	Honeywell Europe	Scottish Power	Elsevier Science	Tesco	VF Corporation	Bosch Group	Deutsche Bank	General Motors
Systems Integrated through Enterprise Application Integration															
Custom Systems	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Packaged Applications					✓	✓	✓	✓	✓			✓	✓	✓	✓
E-business and Electronic Commerce Solutions										✓	✓	✓	✓	✓	✓
Data Vs Process Centric															
Data integration	✓	✓				✓			✓					✓	
Process integration			✓	✓	✓		✓	✓		✓	✓	✓	✓		✓
Barriers to Enterprise Application Integration															
Politics and political impact (e.g. who controls the processes)	✓					✓	✓								
Resistance to change	✓														
No single EAI product solves all integration problems														✓	
No time for training employees on integration technologies														✓	
Extra cost for redesign and change business structure, processes														✓	
Lack of employees with EAI skills						✓	✓							✓	
Cultural issues											✓				
EAI has a high cost						✓									
High complexity in understanding the processes and systems in order to redesign and integrate them							✓								
Earlier approaches on EAI had proved problematic															✓
Complexity of business processes				✓			✓								

Table 2.8: Barriers of Enterprise Application Integration

2.6.1 Classification of EAI Barriers

In this section, application integration barriers are classified according to the model proposed by Shang and Seddon (2000) and are summarised in Table 2.9.

Dimension	Sub-Dimension	Reference
Operational	• Extra cost for redesign and change business structure, processes • EAI has a high cost (initial cost, maintenance etc.)	• Edwards and Newing (2000) • Duke *et al.*(1999)
Managerial	• Lack of employees with EAI skills • Earlier approaches on EAI had proved problematic	• Markus (2000) • Brown (2000)
Strategic	• Resistance to change • Organisations are reluctant to share their data and processes with business partners	• Edwards and Newing (2000) • Kalakota and Robinson (1999)
IT Infrastructure	• No plug and play EAI solutions • No single EAI product solves all integration problems • No single integration technology solves all integration problems • Integration technologies are confusing • Integration solutions are based on a combination of EAI products and integration technologies • Lack of knowledge regarding EAI • High complexity in understanding the processes and systems in order to redesign and integrate them • Lack of enterprise architecture • Lack of common definitions and standards • Existing systems have restrictions regarding their integration capabilities • Lack of documentation especially in the case of custom systems • Many existing systems are complex and incompatible • Some EAI products and technologies are immature	• Linthicum (1999a) • Duke *et al.*(1999) • Ring and Ward-Dutton (1999) • Ruh *et al* (2000) • Linthicum (1999b) • Markus (2000) • Edwards and Newing (2000) • Brown (2000) • Duke *et al.*(1999) • Zahavi (1999) • Zahavi (1999) • Kalakota and Robinson (1999) • Duke *et al.*(1999)
Organisational	• Politics and political impact (e.g. who controls the processes) • Complexity of business processes • Cultural issues • No time for training employees on integration technologies	• Edwards and Newing (2000) • Brown (2000) • Edwards and Newing (2000) • Markus (2000)

Table 2.9: Classifications of Application Integration Barriers

2.7. Costs Related to Enterprise Application Integration Adoption

The introduction of EAI requires organisations to invest considerable amounts of money on their IT infrastructure. These costs include among others the following:

- **Strategic and organisational costs**, which refer to the costs that deal with: (a) business strategy re-design; (b) business process reengineering and, (c) organisational restructuring. Linthicum (1999a) and Brown (2000) report that enterprises turn to application integration for many reasons including gaining competitive advantage. However, to achieve a competitive advantage by focusing on EAI, organisations have to re-design their business strategy. Thus, the cost of redesign the business strategy should take into consideration when introducing EAI. In most cases, the adoption of application integration requires changes to existing business processes. Depending on the mode of integration this requires minor or major business process re-engineering which is an additional cost. Studies have shown that process re-engineering covers more than 60% of the overall EAI project time (Themistocleous and Irani, 2001b; 2002a). This is translated into a significant cost factor. Business process re-engineering requires organisational restructuring to reflect the changes in the processes, which also has an additional cost.

- **Managerial and operational costs** are related with: (a) management efforts; (b) covert resistance; (c) changing employees culture and, (d) employees training. As mentioned in the previous section, the introduction of EAI causes resistance to change and politics conflicts. To address these issues, organisations often have to pay extra money to covert this resistance. In many cases this is achieved through training or organisational restructuring. In addition, the introduction of a new system or software solution requires qualified users. Likewise, the introduction of ERP, organisations have to train their employees when adopting EAI to efficiently support the system.

- **Technical costs.** Technical costs are associated with the: (a) hardware; (b) software; (c) development; (d) maintenance; (e) project management and, (e) consultancy costs. As a typical IT project, the implementation of an EAI solution is associated with the adoption of relevant hardware (e.g. servers, routers, networks) and software. Software costs cover both EAI packages and non-EAI software. For instance network software does not classified as EAI software but it is prerequisite for an EAI solution. Maintenance costs refer to the cost of software licences as well as the cost of maintaining the integrated infrastructure.

2.7.1 Classification of EAI Costs

Many classifications of costs that are related to the adoption of technology were proposed in the literature with Irani and Love (2001), Irani (1998), Irani *et al.* (1998) and Hochstrasser (1992) categorise costs into direct and indirect cost factors. *Direct costs* are financial tangible and are those that can be attributed to the implementation and operation of IT costs. Such costs may include initial hardware and software costs, maintenance costs (e.g. licenses, hardware and software maintenance), system development costs etc. *Indirect costs* are financially tangible/intangible and non-financial in nature and can be divided into indirect human costs and indirect organisational costs. *Indirect human* costs can include employee training, employee motivation, management effort and dedication where indirect organisation costs may include business process reengineering, losses in productivity, strains on organisation resources, organisational restructuring etc. Based on this analysis, Table 2.10 classifies EAI costs.

Dimension	Sub-Dimension	Reference
Direct Costs	▪ Hardware costs ▪ Software costs ▪ Development costs ▪ Maintenance costs ▪ Consultancy costs	▪ Edwards and Newing (2000) ▪ Ruh *et al* (2000) ▪ Duke *et al.*(1999) ▪ *Linthicum* (1999a) ▪ Ring and Ward-Dutton (1999)
Indirect Human Costs	▪ Employees training ▪ Changing employees culture ▪ Management efforts	▪ Markus (2000) ▪ Brown (2000) ▪ Edwards and Newing (2000)
Indirect Organisational Costs	▪ Business Process re-engineering ▪ Organisational restructuring ▪ Covert resistance ▪ Strategy redesign	▪ Kalakota and Robinson (1999) ▪ Edwards and Newing (2000) ▪ Ring and Ward-Dutton (1999) ▪ Brown (2000)

Table 2.10: Classification of EAI costs

2.8. Conclusions

This chapter attempts to review the normative literature to identify research issues. In doing so, the author determines a gap in literature dealing with the absence of theoretical models for EAI adoption. The explanation for this is that enterprise application integration is a new research area. In addition, this chapter reviews the literature on application integration area and discusses in detail the nature of application integration. In doing so, the chapter

- proposes a novel taxonomy for categorising types of EAI;

- classifies the system types that are integrated;
- classifies EAI benefits;
- classifies EAI barriers and
- classifies EAI costs.

Much terminology in the integration area has led to a debate regarding the capabilities of EAI, as each term proposes a different dimension of EAI application. As a result, there is a need to define the dimensions-types of application integration. In doing so, a novel taxonomy is proposed to clarify the confusion around EAI. The taxonomy allows integrators to better navigate, categorise and explain the types of applications (e.g. packaged systems integration) to be integrated. The proposed classification is based on a comprehensive literature review, as well as on published case studies on EAI and separates applications into three main subcategories:

- Intra-organisational EAI;
- Hybrid EAI and
- Inter-organisational EAI.

The first subcategory includes the integration of intra-organisational systems such as packaged and custom solutions. The second subcategory describes the integration of business to consumer applications. The applications of this sub-category are characterised as hybrid, as in some cases these applications function as intra-organisational EAI and in some other as inter-organisational applications. The last sub-category includes business-to-business applications integration and it is further classified according to the degree (loose, tight) of integration. Based on this novel taxonomy, developers can better navigate and understand the integration area and apprehend its capabilities regarding the range of applications that incorporates. Moreover, integrators will better understand the focus of each integration definition in terms of applications, as the taxonomy maps the definitions with the categories of applications.

Although, the proposed taxonomy explains the types of information systems (e.g. packaged systems) that are integrated and categorise EAI (e.g. inter-organisational EAI), it does not describe any classifications of systems types. Thus, the fourth section of this chapter navigates through the types and the permutations of system types that are integrated using EAI. Based on published case studies on application integration area, the author identifies and defines all possible unique permutations of systems types that are used when integrating applications. In doing so, the author addresses a relative void in EAI area, since the classification of system types have not yet been described and identified in EAI literature. The proposed classification of system types could allow researchers and system analysts to apprehend the integration area. In addition, the classification could be used as evaluation criteria when assessing the integration technologies and therefore, help organisations adopting appropriate set of EAI technologies.

The next two sections investigated the EAI benefits and barriers. The author proposes classifications of EAI benefits and barriers. These classifications map benefits (or barriers) against the types of systems that are integrated and the mode of integration that is followed. The data for this classification derive from the analysis of published case studies. This categorisation provides better understanding and helps researchers to analyse the benefits and barriers of EAI. In addition to this categorisation, the EAI benefits and barriers are classified according to model proposed by Shang and Seddon (2000) which separates them into: (a) operational, (b) managerial, (c) strategic, (d) IT infrastructure and, (e) organisational. Thereafter, EAI costs are discussed and classified into: (a) direct costs; (b) indirect human costs and, (c) indirect organisational costs.

Two important research issues came from the literature review presented in this chapter. The first issue is that there is a theoretical gap in the enterprise application integration area regarding its adoption. The second issue is that there is much technological confusion surrounding EAI technologies, which form a significant barrier to its adoption. There is no single integration technology solving all integration problems. As a result, different combinations of integration technologies are needed when integrating applications with organisations need assistance to select appropriate set of technologies. The research issues that are derived from the literature review presented in this chapter are taken into consideration and are addressed in Chapter 3.

Chapter 3: Evaluating and Adopting Integration Technologies

Summary

The previous chapter has concentrated on discussing the nature of application integration and categorising its benefits, barriers and costs. The main research issues that derive from Chapter 2 are that: (a) there is an absence of research and theoretical models that describe the adoption of EAI and, (b) integration technologies form a barrier to EAI adoption. Integration technologies are a barrier to EAI adoption since a diversity of technologies and EAI products exist in the marketplace. These technologies differ in the type of the solution they provide and, focus at various levels of integration (e.g. data, objects etc). However, none of these technologies claim to be a panacea to overcoming all integration problems. There is therefore, often a need to combine integration technologies as a means of *piecing* together applications.

The aim of this chapter is twofold: (a) to attempt to clarify the technological confusion surrounding EAI area and, (b) to conceptualise a model for the adoption and evaluation of EAI. The author addresses the former by proposing an evaluation framework for the assessment of integration technologies. The framework is based on a comprehensive set of evaluation criteria that clarify much of the confusion surrounding EAI. Thereafter, the evaluation framework is used as part of a novel conceptual model that is proposed for EAI adoption. The proposed model attempts to contribute in EAI adoption area, as it describes a number of factors that influence EAI adoption.

The first section of this chapter explains in technical terms how application integration is achieved. This provides relevant background to the following sections in which the integration requirements and technologies are described. The second section categorises

common integration requirements of information systems based on the classifications of system types defined in Chapter 2. In addition, integration requirements that derive from the literature and are related to an integrated solution are also summarised. These requirements are used in section 3.4 as evaluation criteria and are taken into consideration for the development of the proposed evaluation framework.

The third section describes integration technologies that can be used to meet the integration requirements and unify applications. The description of such EAI technologies is necessary, since the proposed framework focuses on evaluating integration technologies. Thus, such a description allows the reader to better understand the evaluation framework. Section 3.4 introduces then, a novel evaluation framework for assessing EAI technologies. The evaluation framework is based on criteria that are derived from a comprehensive literature review. The author divides the evaluation criteria into: (a) integration requirements (e.g. flexibility, portability); (b) the application elements that are integrated (e.g. data, objects); (c) the integration layers (e.g. transportation, transformation) and, (d) the classification of system types that are integrated (e.g. custom-to-custom EAI). The proposed evaluation framework clarifies the differences among technologies and will support integrators during the selection of an appropriate permutation of integration technologies (for the purpose of integration). At a technical level, the evaluation framework contributes towards a better understanding of the capabilities of each technology, and highlights combinations of integration solutions. Based on this framework and the evaluation results, possible permutations of technologies that can address the integration of information systems are highlighted. Such a framework can be used as a decision making tool by IT departments or business analysts when taking decisions to integrate enterprise and cross enterprise applications. Therefore, the proposed novel framework can be considered as a factor that influences the adoption of enterprise application integration.

Section 3.6 proposes a novel conceptual model for the adoption of EAI. The model takes into consideration and extents previous approaches to EDI adoption, since: (a) EDI is considered as an integration technology and, (b) there is an absence of other models in the normative literature regarding EAI adoption. The proposed model makes novel contribution at two levels. First, at the conceptual level, as it incorporates factors identified separately in previous studies as influencing the adoption of EDI and other technologies (e.g. Iacovou et *al.* (1995)). These factors are used for the development of a consistent model for the adoption of application integration. Secondly, the concepts of the proposed model can be used for the adoption of inter-organisational information systems. The proposed model can be used as a decision-making tool and, supports management when taking decisions regarding the adoption of EAI. Additionally, it can be used by researchers to analyse and understand the adoption of application integration.

3.1 Application Integration in Technical Terms

The integration of IS applications is an obstacle to many businesses, as they consist of independent systems, which in some cases can not communicate with one another. These autonomous and in many cases heterogeneous systems are historically not designed to collaborate with other applications, since their focus was to overcome specific point problems (Swenson and Cassidy, 1993). As a result, departments within the same organisation have tended to develop their systems independently and without any enterprise wide co-ordination. This may result in a lack of enterprise architecture, common definitions, structures, protocols and business concepts (Duke *et al.*, 1999). This is further complicated by information systems being based on a plethora of different standards, computing languages, platforms and operating systems, which cause various integration problems such as incompatibility (Kim and Umanath, 1999; Makey, 1998). Wijegunarate and Fernandez (1998) suggest that there is also the complexity of existing information systems, which in many cases have fixed and rigid structures for messages, interfaces and databases. Moreover, there is a lack of documentation, especially as legacy systems have often-important technical information missing. The reason for this is that, legacy systems exist in organisations for more than 35 years and their technical documentation was either not created or lost during the years. As a result the integration of applications at an enterprise and cross enterprise level is a complex task.

Traditionally, most integration projects incorporate applications by developing manual point-to-point connections. Programmers write low-level communication code between two applications to exchange messages and data. However, such approachs have led to *applications spaghetti*, which increases the complexity of the integration solution as the number of interconnected applications rises. Themistocleous *et al.*(2000) suggest that for *x* applications a total of:

$$x*(x-1)/2$$

connections are required, to piece together *all x* applications. This means that for 10 applications, 45 connections are needed. Ring and Ward-Dutton, (1999), Stonebraker (1999) and Pender (2000) support this finding, since they report that when point-to-point connections are used to integrate IS, all applications are required to be pieced together. Clearly, maintenance costs are an issue, with IT becoming ineffective to maintain these interconnected applications. Moreover, interconnectivity has other problems since point-to-point connections have an *invasive nature*, which requires changes to applications (Duke *et al.*, 1999; Serain, 1999; Themistocleous and Irani, 2002b; Wijegunarate and Fernandez, 1998). Thus, new subroutines that support interconnections by mapping all interconnected applications are added and therefore, application code is extended. After interconnecting applications, if a system requires changes, all interconnected applications have also to be altered. As the number of applications and connections between them proliferate, an organisation ends up with a non-flexible, unmanageable jumble of code holding the business system together.

Application integration addresses integration problems more effectively by developing a central integration infrastructure. In doing so, point-to-point interconnections are eliminated, since each application is connected with an integration infrastructure. In many cases, the integration infrastructure follows a hub and spoke communication mechanism (Bernus *et al.*, 1996), which is often based on a message broker (Ruh *et al.*, 2000; Themistocleous and Irani, 2002b). In EAI solutions, when an application requires changes, the rest of the system is rarely affected, as it is not interconnected with the application that requires changes. Therefore, only the application that requires changes and its connection to the central integration infrastructure are altered.

Such a solution increases flexibility and maintainability since changes are limited, and interconnections are fewer in number. Moreover, many of the integration technologies being used to incorporate applications do not have an invasive nature, which also increases flexibility and manageability. Figure 3.1 illustrates the differences in the number of connections when traditional integration approaches and application integration are adopted.

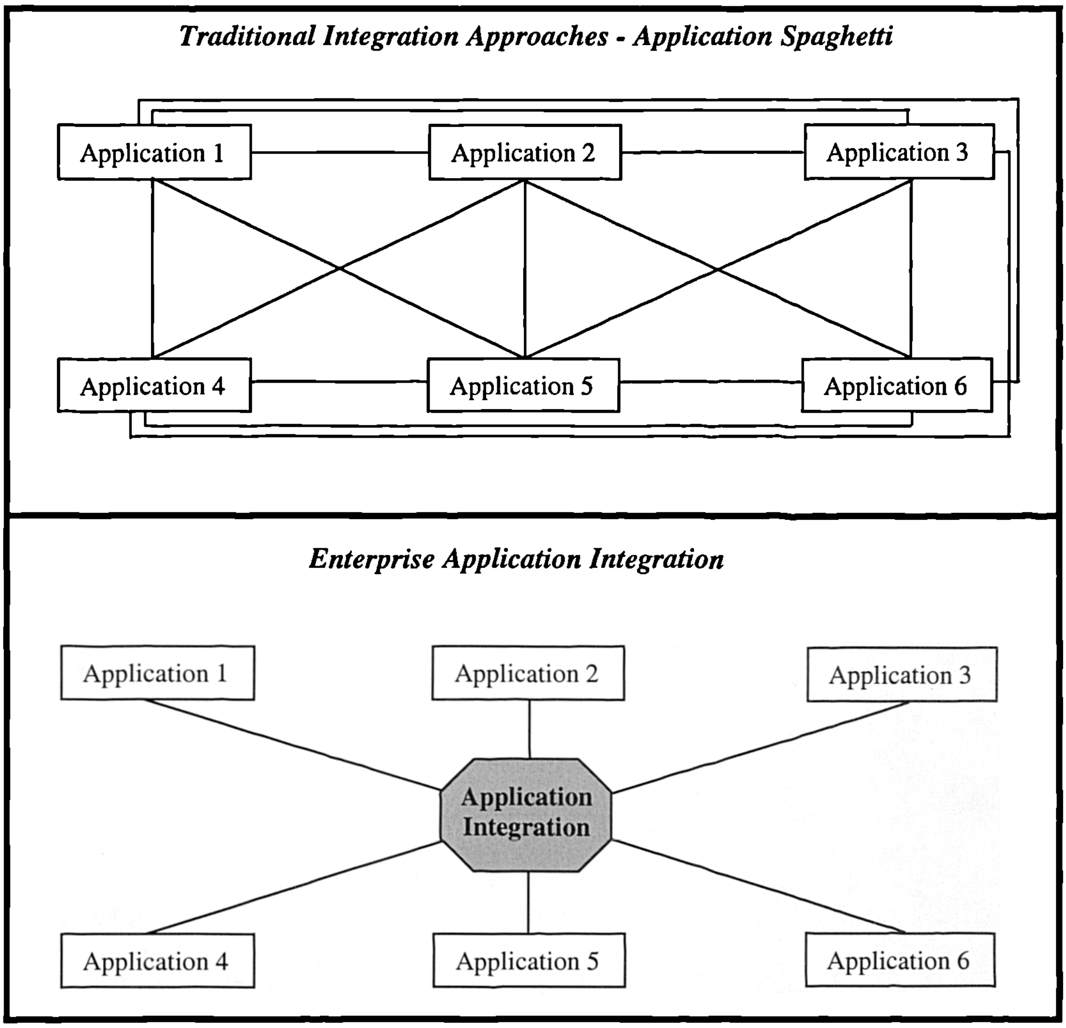

Figure 3.1: Application Spaghetti Vs Enterprise Application Integration

Numerous approaches were proposed in literature to describe application integration. Duke *et al.*(1999) suggest that a solution based on application integration involves the transportation and transformation of information between one or more applications. It also supports: (a) the timing and sequencing rules that govern when the transportation and transformation takes place and, (b) the integrity constrains that determine the success or failure of the integration. Themistocleous *et al.*(2000) evaluate and summarise integration approaches that were found in the EAI literature and, report these in Appendix A. From a technical perspective Themistocleous *et al.*(2000) propose that EAI is achieved at three integration layers namely:

- **Transportation layer**, which transfers the information from source application to the integration infrastructure and from the latter to the target application.
- **Transformation layer** that translates the information from source application format to target system structure.
- **Process automation layer**, which integrates the business processes and controls the integration mechanism. This is illustrated at Figure 3.2

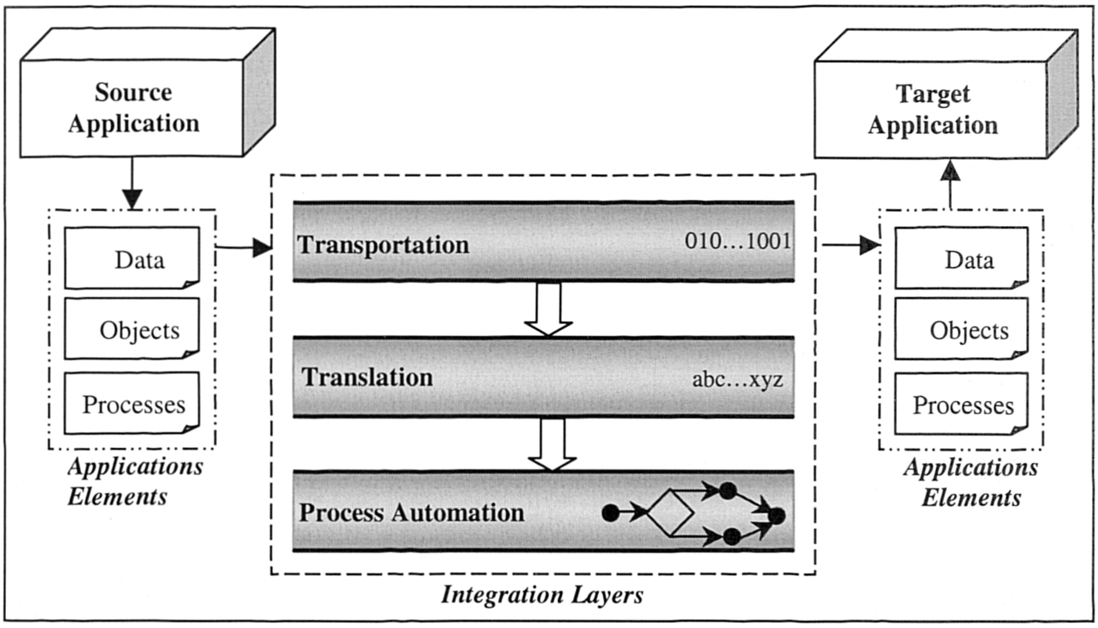

Figure 3.2: Integration Layers and Application Elements

Figure 3.2 presents the incorporation of two information systems (source and target application) when EAI is used. Both source and target applications are based on the classification of system types as identified in Chapter 2. Source and target applications are integrated by exchanging their *application elements, which include data, objects and processes*. Application elements are transferred from source application to the target through the integration infrastructure using the *transportation layer*. Meanwhile, application elements are translated from source application structure to target application format using the *translation layer*. The reason for using the translation layer is that source and target applications are not based on the same structures (e.g. data structure) or platform. Thus, translation is needed to transform data into compatible format for target application. At a higher level, elements that are used for the integration of processes or the integrity of information (e.g. services, business logic, rules, constrains) are transferred to *process automation layer*. These elements are used by process automation layer to audit integration tasks, incorporating and automating business processes and triggering events. For instance a typical task of this layer could be the following scenario: translate retailer's (source application) stocks data using the translation layer. When the retailer's stocks availability is equal to limit (e.g. stocks = x units of product P) then notify supplier (target application) and order z = y-x units of product P (where y is the maximum agreed quantity of product P).

Based on this analysis, the author observes that *technologies that efficiently support the integration of: application elements; (b) integration layers; (c) classifications of system types being integrated and, (d) integration requirements of source and target applications, can be used to piece information systems together.* This can be rephrased into that: (a) application elements; (b) integration layers; (c) classification of system types and, (d) integration requirements can be used as evaluation criteria when assessing integration technologies. The reason for this is that they efficiently describe the applicability of integration technologies. This observation is further analysed and explored within the following sections.

3.2. Integration Requirements

Maciaszek (2001), Britton and Doake (1996) and Robertson and Robertson (1999) suggest that it is not feasible to propose a generalised model that specifies the requirements of all information systems since, each system operates in its own environment and it is based on different standards and platforms. Also, in the integration literature, Lambert *et al.* (1998) and Lloyd *et al.* (1999) suggest that it is difficult to specify and generalise the integration requirements of information systems. Even in those cases where many organisations have adopted the same packaged system (e.g. the financial module (FI) of SAP system, version R/3 for windows NT 4.0) it is difficult to identify all integration requirements as each organisation has customised this system according to its own business needs.

Nonetheless, Duke *et al.* (1999) classify common integration requirements of information systems by focusing on three generations of information systems namely: (a) *Stovepipes* which include systems that were developed before 1980; (b) *Tunnels* that refer to applications developed in 80s and 90s and (c) *Blobs* that refer to e-business solutions. Clearly, these three generations of information systems refer to custom, packaged and e-business solutions respectively. Based on Duke *et al.* (1999) approach, the author supports that although information systems use disparate computing languages, platforms and standards, they have a number of common characteristics. For example custom systems have a monolithic nature, packaged applications follow a 3-tier model and e-business solutions are distributed componentised systems. Thus, their *common integration requirements* can be identified according to these characteristics.

The author of this dissertation attempts to expand the work of Duke *et al.* (1999). In doing so, specifying the common integration requirements using the classifications of the system types defined in Chapter 2. Such analysis that focuses on common requirements of systems being integrated will highlight points of integration for each system type. This will help researchers and practitioners to better understand and analyse integration problems and will support the selection of appropriate set of integration technologies.

The common integration requirements of system types are presented below:

- **Custom applications** like legacy systems operate on non-windows platforms and the majority of them run on mainframe environments (Bennett, 1995; Brodie and Stonebraker, 1995; Butler Group, 1998). According to Robertson (1997) and Noffsinger *et al.* (1998) they have a monolithic nature and thus, their data, interfaces, business logic, rules and constrains are built together. As a result, custom systems were not designed to collaborate with other applications and therefore, have limited points of access (integration). In most cases, user interfaces are the only point of access. Therefore, integration technologies that extract data, business rules and logic from user interfaces and support mainframes can be used to piece together custom-to-custom applications (Linthicum, 1999b; Zahavi, 1999). Nevertheless, in those limited cases that custom systems use databases to store their data, database oriented technologies that support mainframes can be adopted to extract data from custom applications.

- **Packaged applications** such as ERP solutions were not designed to incorporate other autonomous applications, a diversity of technologies, techniques and tools can be used to piece them together (Loos, 2000; Meier *et al.*, 2000; O'Leary, 2000). Packaged systems are based on a client server model and run on both mainframes and non- mainframes environments. The latest versions of packaged systems run on various platforms such as the windows operating systems, Unix, Linux etc. Packaged systems have two main points of integration: (a) the Application Programming Interfaces (APIs) and, (b) databases (O'Leary, 2000; Ring and Ward-Dutton, 1999; Sherlund *et al.*, 1999). Therefore, technologies that support the integration of APIs and databases can be used to unify packaged systems with an integration infrastructure. Technologies that are used for packaged application integration should also support mainframes and/or non-mainframes environments for the reasons explained above.

- **E-business solutions** have been designed to collaborate with existing information systems and infrastructures. Most e-business applications are distributed, componentised and run more on windows based environments (Linthicum, 2000). E-business solutions store their data into databases and support java enabled and/or non-java enabled platforms (Morgenthal and La Forge, 2000). For many organisations real-time integration is a significant requirement for the functionality of their e-business applications. Other integration requirements include extracting and inputting data, objects, business rules and logic. Data are extracting and inputting from/to databases where objects, business rules and logic from APIs.

Table 3.1 summarises the common integration requirements for the classifications of system types.

	Classifications of System Types						
	Custom to Custom	Custom to Packaged	Custom to e-business	Packaged to Packaged	Packaged to ebusiness	Ebusiness to ebusiness	Custom to Packaged to ebusiness
Integration Requirements							
Extract and input data from/to databases that run on mainframes.	✓	✓	✓	✓	✓		✓
Extract and input data from/to databases that run on non-mainframes (e.g. windows based) environments.		✓	✓	✓	✓	✓	✓
Extract and input data from applications that run on mainframes using their user interfaces.	✓	✓	✓				✓
Extract and input objects from/to user interfaces that run on mainframes.	✓	✓	✓				✓
Extract and input objects from/to APIs that run on non-mainframes environments.		✓	✓	✓	✓	✓	✓
Extract and input business rules and logic from/to applications that run on mainframes using their user interfaces.	✓		✓				✓
Extract and input business rules and logic from/to applications using their APIs		✓	✓	✓	✓	✓	✓
Real time integration			✓		✓	✓	✓

Table 3.1: Integration Requirements based on the Classifications of System Types

Apart from these common requirements for system types, Zahavi (1999), Puschmann and Alt (2001) and Themistocleous and Irani (2002b) support that organisations also focus on another set of integration requirements when adopting EAI technology. This second set describes characteristics (e.g. reusability, portability) for the overall integration solution or the technologies being used. Based on an extensive review of the literature and published case studies on EAI, the author summarises the more common used requirements. Table 3.2 presents these integration requirements. Requirements summarised in Table 3.2 are important since, developers take them into consideration when integrating their IS. Therefore, such requirements can be used as evaluation criteria when selecting an integration technology.

Integration Requirements	Description
Maintainability	Britton and Doake (1996) suggest that maintainability is an important characteristic of software technologies. It refers to the capability of software applications to allow changes without causing problems to other components or systems. In integration area, technologies should lead to the development of solutions that could be easily maintained.
Flexibility	Mandelbaum and Buzacott, (1990) report that flexibility makes a system or a process able to respond to change in the systems environment. Knoll and Jarvenpaa (1994) identify three types of flexibility: flexibility in functionality, flexibility in modification, and flexibility in use. The first two types of flexibility describe the capability of rapid adjustments with minimal effort and the capability to operate well in many different environments (Knoll and Jarvenpaa, 1994). In the context of integration technologies, flexibility supports both flexibility in functionality and modification as well as Mandelbaum and Buzacott (1990) definition.
Scalability	Scalability describes the ability of an information system to provide high performance as greater demands are placed upon it, through the addition of extra computing power (Linthicum, 1999a).
Portability	Portability allows a software solution developed for one platform to run on an entirely different platform (Mooney, 1995; Rowley, 1996). Portability is closely related to the concept of standards and plays a significant role in the cost-effectiveness of information systems (Rowley, 1996).
Reusability	Reusability refers to the capability of using existing components or software solutions to build new applications (Krueger, 1992; Mooney, 1995). Reusability, has an important role in application integration as it reduces the implementation time and cost. It results in more flexible, manageable and maintainable systems (Mowbray and Zahavi, 1995).
Maturity	Maturity shows whether an integration technology is mature or not. The more mature an integration technology, the better it is. The reason for this is that analysts and developers trust more mature technologies than immature one.
Complexity	Complexity describes whether an integration technology leads to complex or simple solutions. Often, complex integration solutions are not preferred as they increase development and maintenance costs.
Non-Invasive	Many organisations seek for non-invasive integration technologies and solutions. The reason for this is that invasive technologies (like RPC) extend the code of interconnected applications by adding new modules that support integration efforts. This, result in non flexible, maintainable and portable solutions since, a change in one application will affect all the interconnected systems. As a result, the less changes required the better value achieved for integration as cost, effort and complexity are eliminated and flexibility and maintainability are increased (Morgenthal and La Forge, 2000).
Performance	In some cases integration technologies achieve integration but the performance of the overall solution could be low. As a result, this requirement seeks to describe whether the performance of an integration solution is low or not. The higher the performance the better it is.
Real Time	Real time requirement refers to the capability of integration technologies to support transactions that require up to the second data latency (Linthicum, 1999a). Data latency defines how current the information needs to be. As mentioned in previous paragraphs real time integration is important for e-business applications

Table 3.2: Integration Requirements based on Characteristics of EAI Technologies

3.3. Integration Technologies

This section presents the integration technologies that can be used to incorporate the permutations of system types (e.g. custom-to-packaged) defined in section 2.4 and support integration requirements.

Application integration is based on a plethora of integration products and technologies to efficiently support the incorporation of information systems. However, *no single integration technology addresses all integration problems*. Ruh *et al.*(2000) classify these technologies in five categories namely:

(a) database oriented middleware;
(b) message oriented technologies;
(c) Transaction based technologies;
(d) distributed object technologies (DOT) and
(e) interface oriented technologies.

These categories of integration technologies are summarised below since the evaluation framework that is proposed in section 3.4 focuses on the assessment of these technologies. Therefore, a summary of integration technologies will allow the reader to better understand the evaluation framework and the interpretation of evaluation (see section 3.5).

3.3.1 Database Oriented Middleware

Database oriented middleware is the software that connects an application to a database using a well-defined Application Programming Interface (API). It is fundamental for application integration, as most data are stored in databases that are accessible using this category of middleware technology. Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC) have become popular standard mechanisms that support access to distributed databases and are described below:

- **Open Database Connectivity** provides a well defined and database independent API that simplifies database access from both windows and non windows operating systems.

It exposes a single API to facilitate access to a database and then determines the appropriate ODBC driver to support the translation of data from the source database to the target. ODBC focuses more on relational databases than other types of databases such as multidimensional, object-oriented or hierarchical databases. Brown (2000) says it should be considered when operating in a multi-database environment that requires access to several databases from the same application or integration server (e.g. message broker, application server).

- **Java's Database Connectivity** functionality is similar to ODBC. It provides access to most relational databases from Java-enabled applications and environments. It also provides database access for many EAI enabled products such as Application Servers, Message Brokers and Message Oriented Middleware (MOM). Although both JDBC and ODBC simplify database access, they do not solve the integration problem since the data must be distributed, identified, classified, and altered to reach the target application. Hence, both JDBC and ODBC (and database oriented middleware in general) have to collaborate with other technologies to achieve this functionality.

3.3.2 Message Based Technologies

This category of integration technology manages the distribution of messages from one application to the other. Messages include data, objects and components that are sent from source application to target. Messaged based integration technologies include: (a) Remote Procedure Calls; (b) Message Oriented Middleware (MOM); (c) Message Brokers and (d) Extensible Markup Language (XML). These categories are described below.

- **A Remote Procedure Call** (RPC) is a mechanism that makes message-passing look like procedure call (Bernstein, 1990). Linthicum (1999b) reports that RPCs hide the complexities of operating systems and networks through a function call. It focuses on the integration of the procedures of distributed applications across a network (Ruh *et al.*, 2000) and based on a synchronous (Birrell and Nelson, 1984; Linthicum, 1999b) Point-to-Point communication model (Edwards and Newing, 2000). The invasive nature of RPCs is a significant drawback and leads to high maintenance costs and complexity

(Linthicum, 1999b). In addition, RPCs requires high-speed networks, high processing power, and a high level of detail technical input and it is complex to replicate the results (Edwards and Newing, 2000). RPC is the only type of Middleware that declines (Ruh *et al.*, 2000). The reason for this is that RPC follows an old style of programming (procedures) that is decline. Furthermore, other integration technologies such as MOM and DOT (CORBA, DCOM) are more functional and provide similar functionality to RPCs (e.g. DOT products support synchronous communications) (Linthicum, 1999a; 1999b; Ruh *et al.*, 2000).

- **Message Oriented Middleware** (MOM) piece together applications using messages as the method of integration, and supports the development, manipulation, storage and communication of messages. Data that are extracted from a source application are included in a message, which is transmitted to target applications through MOM. Message oriented middleware is based on a *point to point asynchronous messaging mechanism* for the communication of applications, and requires altering the source and target application, which increases maintenance costs and complexity. In addition, MOM can not be effective for objects/component integration, as MOM messages are not as easily visible as interfaces.

- **Message brokers** have an important role in applications integration since they result in flexible, non-invasive and easier to maintain integration mechanism. Message brokers move messages from one application to another by changing or translating the format of messages thus, supporting the needs of the target applications. The purpose of message brokers are to integrate multiple business processes and applications using adapters (Edwards and Newing, 2000; Linthicum, 2000). Message brokers support the transformation of data and messages, message filtering and routing. In addition, they provide business rules processing capabilities, hosting business functions, message translation engines and bridges to many different platforms and applications. Message brokers lack abstraction and object-oriented capabilities since they are based on messaging. According to Zahavi (1999)s a permutation of message brokers with component capabilities will offer a more robust EAI infrastructure and thus, better address integration problems.

- **Extensible Markup Language (XML)** is an internet based meta-language that provides a standard mechanism for data exchange between applications and companies (Cingil *et al.*, 2000; Ring and Ward-Dutton, 1999). XML can support EAI in two ways; firstly it can be used as a description language to specify resource interfaces. All messages sent by resource interfaces to a transformation service can be self-describing through XML. As a result, it simplifies the implementation of message validation service. Secondly, it can be used as a metadata standard for object integration (Ring and Ward-Dutton, 1999). However, XML is not a panacea for EAI, as it does not address issues such as process integration (Smith and Poutler, 1999). Moreover, during an XML interchange, redundant and unnecessary information (e.g. tags, metadata) is transmitted. XML may cause interoperability problems as users define their own tags (Cingil *et al.*, 2000). Recently, many industries attempt to overcome this problem by developing their standards-tags.

3.3.3 Transaction Based Technologies

Transaction based technologies function around the notion of transaction, which is defined by Ruh *et al.* (2000, pp. 108-109) as *"a single unit of work to support one or more business functions that must be completed in a single action to achieve a business purpose. If not all required business functions can be completed, then the transaction not be completed."* Transaction Process Monitors (TP Monitors) and Application servers are the main technologies of this category and are summarized below:

- **A Transaction Process (TP) Monitor** is a mechanism that co-ordinates the flow of transaction requests between terminals, or other devices and applications (Bernstein, 1990). Also, TP Monitors facilitate the communication between two or more applications and provide a location for application logic. Application logic is encapsulated within a transaction. If problems occur during the transaction, then the transaction rolls back[1]. TP Monitors provide tools to ensure the integrity of complex business processes by providing atomicity, consistency, isolation and durability of transaction and support features such as auto restart, error logging and replication, fail-over and rollback to eliminate failure (Linthicum, 1999a; Ruh *et al.*, 2000; Zahavi, 1999). Recently, there has been a trend for merging TP Monitors and DOT (e.g. COM, CORBA etc) as the latter provide TP Monitors services (Linthicum, 1999a). Moreover, MOM technology has also incorporated TP Monitors functionality (Ruh *et al.*, 2000). TP Monitors integrate services that make them accessible through a simplified API (Bernstein, 1996) and can connect with databases, message queues and other applications.

- **Application Servers** function around the notion of transactional components. They support sharing and processing of application logic and provide connections to back-end resources such as ERP systems. In addition many vendors are attempting to incorporate functionality from message brokers to application servers (e.g. intelligent routing, transformation, messaging), and are becoming more functional and flexible. They provide the infrastructure support for executing distributed applications with technology integration capabilities (Duke *et al.*, 1999). Application servers can communicate with distributed object technologies and increase their functionality. However, they do not support content or message transformation services without much of programming.

3.3.4 Distributed Object Technologies (DOT)

Distributed object technologies support the development of object-oriented interfaces to existing or new applications that are accessible from any other application. Distributed objects technology allows the sharing of data, application logic and provides a central clearinghouse for enterprise information. Common Object Request Broker Architecture (CORBA), Component Object Model (COM), Distributed Component Object Model (DCOM) and Enterprise Java Beans (EJB) are the main object technologies that are used to achieve integration.

- **CORBA** was designed as an open standard to provide integration and interoperability in distributed systems. It is available on more than twenty-five different platforms and is suitable for applications that exist in a heterogeneous environment, especially when an organisation supports a variety of systems and platforms (e.g. UNIX, mainframes etc).

[1] Roll back means that if an error occurs during a transaction, the system restores its data to its

CORBA components can be used to support multiple programming languages, with many ERP vendors (e.g SAP) integrating their packages with CORBA or Component Object Model (COM). CORBA supports legacy integration, platform independence and location transparency. It is flexible and provides bindings to different languages and platforms.

- **Component Object Model (COM)** is an object standard provided for Windows platforms (Rosen, 1998). Like CORBA, COM provides the rules that developers should follow when creating COM-compliant distributed objects. COM and DCOM support multiple languages but they differ from CORBA in their approach to support these languages.

- **Distributed Component Object Model (DCOM)** is part of the Windows operating system (Windows 98, Windows NT, and Windows 2000), and allows COM enabled application to locate and use remote COM-enabled Object Request Broker (ORB) and find and invoke the service it require (Linthicum, 1999a; Zahavi, 1999). DCOM is compatible with existing COM-enabled development. Similarly to CORBA, DCOM supports more synchronous communication models but, it is immature in development to support asynchronous types of communications.

- **Enterprise Java Beans (EJB)** define a model for the development of reusable Java server components and are similar to COM and CORBA architectures. They support multi-tier distributed object application development, and provide a set of enterprise component interfaces (APIs) (Morgenthal, 1998; Ruh *et al.*, 2000). EJB component model can support asynchronous communications and publish/subscribe services. EJB are communication protocol independent and provides portability (Andrew, 1998).

3.3.5 Interface Based Technologies

Interface based integration requires the specification of well-defined interfaces that describe the actions that an application can perform. Interfaces are means by which users, application

---

previous status

or software components can interact with a given application or component. Interfaces are classified into user interface-screen wrappers, Application Programming Interfaces (APIs) and adapters (or connectors) and are presented below:

- **Screen wrappers.** In many applications such as legacy systems, the user interface is the only available mechanism to access data, logic and processes (Andrew, 1998; Van Den Heuvel *et al.*, 1999). Screen wrappers (or scrappers) are used to encapsulate (extract) data from user interfaces and transform them into raw data (screens as data) or objects (screens as objects). In doing so, they use mapping techniques to map the user's interface information to raw data or objects. Wrappers also expose interfaces over legacy transactions and provide metadata descriptions of legacy systems (Robertson, 1997). The screens-as-objects method requires the translation of data extracted from user interface to an application object (e.g. CORBA, COM or Java) which requires adding the application methods needed to interact with data. As a result, user interface information is transformed into a set of objects that can be processed by DOT. Thus, the screens-as-objects method is more appropriate for AI, since they can also support component based applications.

- **Application Programming Interfaces** (API) is a mechanism provided by an application to access its functionality or data (Ruh *et al.*, 2000). APIs can communicate with DOT such as CORBA and DCOM, and are portable to other applications. The majority of ERP systems use APIs to facilitate the communication and integration of these applications. Therefore, ERP systems are attempting to function like distributed object solutions by supporting APIs that provide access to their data, objects, services and processes. APIs can be used for accessing data in real time, and support reusability.

- **Adapters** are a set of libraries that map differences between two distinct interfaces, hide the complexities of interfaces and perform the extraction, translation and input steps of integration. Adapters can be used as translators that plug into application APIs to present standardise data or message interfaces to the connectivity transformation and process management services of an EAI solution. Adapters offer more capabilities than APIs, and support transformation of data into formats that are acceptable to the target application (Ring and Ward-Dutton, 1999). Adapters add more value to EAI solutions by supplying metadata that describes resource behaviour to developers (Ruh *et al.*, 2000).

3.4 Novel Evaluation Framework for Evaluating Integration Technologies

Previous sections of this dissertation (see sections 2.6 and 2.8) reported that there is a confusion surrounding EAI and integration marketplace. In addressing this issue, the author supports that the development of an evaluation framework will support decision-making when organisations select integration technologies. Thus, the evaluation framework will clarify much of the confusion. This section presents a novel evaluation framework that has been developed for this purpose (evaluation of integration technologies). The novelty of the framework focuses on the combination of criteria that efficiently describe the application integration area. As it has been observed in section 3.1, application integration could be efficiently supported by technologies that focus on the integration of: (a) applications elements; (b) integration layers; (c) classifications of system types and, (d) integration requirements of systems being integrated. All these criteria have been discussed in sections 3.1 and 3.2 and are summarised in Table 3.3.

Evaluation Criteria	
Application Elements	**Integration Layers**
▪ Data ▪ Objects ▪ Processes	▪ Transportation Layer ▪ Transformation Layer ▪ Process Automation Layer
Classification of System Types	**Integration Requirements**
▪ Custom-to-Custom ▪ Custom-to-Packaged ▪ Custom-to-e-business ▪ Packaged-to-packaged ▪ Packaged-to-e-business ▪ Custom-to-Packaged-e-business	▪ Maintainability ▪ Flexibility ▪ Scalability ▪ Portability ▪ Reusability ▪ Maturity ▪ Complexity ▪ Non-invasive ▪ Performance ▪ Real-Time ▪ Mainframe compatible ▪ Non-Mainframe compatible

Table 3.3: Evaluation Criteria

- Table 3.3 incorporates the integration requirements of Tables 3.1 and 3.2. Nonetheless, the integration requirements of Table 3.1 have been restructured so that it better supports the whole evaluation framework. For instance, the requirements of Table 3.1 focus on the incorporation of integration elements such as data, objects and processes (e.g. business logic, rules). These requirements indicate that the incorporation of integration elements should support *mainframe* and/or *non-mainframe environments*. Moreover, the same requirements demand technologies to support the integration of databases, APIs and user interfaces when applications run over *mainframe* or/and *non-mainframe environments*. Therefore, the criteria that refer to mainframe and non-mainframe compatibility are incorporated with the other criteria summarised in Table 3.2.

The ranking of integration technologies follows a low (○), medium (◒), high (●) scale of ranking similar to the scale used by Miles and Huberman (1994). In addition, two other symbols are used for ranking. As EAI is an emerging research area, literature is narrow especially from the perspective of evaluating integration technologies. In many cases EAI literature does not make a distinction between whether a technology may receive one value or another (e.g. low or high). However, many of references in EAI (Linthicum, 1999b; Morgenthal and La Forge, 2000; Zahavi, 1999) report that a technology either does not support or supports a specific criterion. As a result, the values: (a) supports or satisfies (✓) and, (b) does not support (✖) are also adopted for ranking. Moreover, the value null (–) indicates that there is no available information regarding an integration technology.

3.4.1 Evaluation of Integration Technologies

This section assesses integration technologies based on the proposed novel evaluation framework. Such evaluation is important since, it attempts to clarify much of the confusion surrounding integration area. The proposed framework highlights possible combinations of EAI technologies that can be used for integrating system types. Therefore, the framework can be used as a decision-making instrument and to support organisations when adopting their applications.

Table 3.3 depicts the evaluation of integration technologies based on the criteria discussed in sections 3.1, 3.2 and 3.4 and summarised in Table 3.3. The author evaluates integration technologies using normative literature and, concludes that no single technology satisfies all evaluation criteria. Therefore, this suggests there is no single technology addressing all integration problems. The evaluation is synopsised in the following paragraphs based on the categories of integration technologies. Such analysis, clarifies the differences among the categories of integration technologies. The main evaluation findings are now summarised.

Database Oriented Technologies: ODBC and JDBC can efficiently support the integration of data by extracting and inputting data into databases and data files. They can also support custom, packaged and e-business application integration. Although both of these technologies can translate data, they can not transfer them to the target application. Therefore, ODBC and JDBC have to collaborate with other technologies to send and receive data. In addition, data oriented technologies do not support process and objects/components integration layers. ODBC and JDBC are mature technologies that provide high performance, have low complexity and support portability and reusability of data. ODBC supports both mainframe and non-mainframe environments where JDBC focuses more on windows oriented platforms.

Message Oriented Technologies: Linthicum(1999a) considers message brokers as the preferable integration technology since, they address integration problems at all integration layers and provide almost all the required characteristics. Although, they were not designed to facilitate real time environments, they can be configured to support them. Message brokers have low maturity and can not be used for objects integration.

Category of Integration Technologies	Integration Technologies	Evaluation Criteria: Integration Requirements												Applications Elements			Integration Layers			Classifications of System Types						
		Maintainability	Flexibility	Scalability	Portability	Reusability	Maturity	Complexity	Non-invasive	Performance	Real Time	Mainframe Compatible	Non-Mainframe Compatible	Data	Objects	Process	Transportation	Translation	Process Automation	Custom to Custom	Custom to Packaged	Custom to e-business	Packaged to Packaged	Packaged to e-business	E-business to e-business	Custom to Packaged to e-business
Database Oriented Middleware	ODBC	-	-	-	✓	✓	●	○	-	●	-	○	●	●	×	×	×	✓	×	◒	◒	◒	●	●	●	●
	JDBC	-	-	-	✓	✓	●	○	-	●	-	×	◒	●	×	×	×	✓	×	○	◒	◒	◒	◒	●	◒
Message Oriented Technologies	RPC	○	×	×	-	✓	●	●	×	○	✓	●	●	✓	×	×	●	×	×	○	○	○	×	×	×	○
	MOM	○	◒	○	×	×	◒	●	×	◒	×	●	●	✓	○	×	●	×	×	◒	◒	◒	○	○	×	○
	Message Broker	●	●	●	●	●	○	○	✓	●	○	●	●	✓	○	✓	●	●	●	●	●	●	●	●	●	●
	XML	◒	●	✓	●	-	◒	○	○	◒	✓	×	◒	✓	✓	×	●	●	×	○	○	●	●	●	●	●
Transaction Based Technologies	TPM	-	○	●	●	-	◒	●	×	●	-	◒	◒	✓	×	×	✓	○	×	◒	○	○	○	○	○	○
	Application Serves	◒	◒	●	●	◒	○	◒	×	◒	-	-	◒	✓	✓	×	✓	×	×	×	◒	◒	◒	●	●	◒
Distributed Object Technologies	CORBA	◒	✓	◒	◒	●	○	-	×	◒	-	-	●	✓	●	-	✓	✓	-	◒	◒	◒	●	●	●	◒
	DCOM / COM	◒	✓	◒	○	●	○	○	×	◒	✓	×	◒	✓	●	-	✓	-	-	○	◒	◒	●	●	●	◒
	EJB	◒	✓	◒	◒	✓	○	-	×	○	✓	×	◒	✓	●	-	✓	-	-	○	○	◒	◒	●	●	◒
Interface Oriented Technologies	Screen wrapper	○	×	◒	-	✓	◒	●	✓	○	-	●	×	✓	✓	×	×	✓	×	●	●	●	○	○	○	●
	APIs	◒	✓	-	✓	✓	◒	●	✓	-	✓	×	●	✓	✓	-	-	✓	-	-	◒	×	●	●	●	●
	Adapters	●	✓	✓	✓	✓	○	○	✓	-	-	-	●	✓	✓	×	×	✓	×	-	◒	◒	●	●	●	●

Table 3.4: Novel Evaluation Framework for the Assessment of Integration Technologies

In contrast, XML is more mature than message brokers, supports objects integration and address more efficiently real time issues. XML addresses data, packaged and e-business integration and can be used for the transportation and translation of data. However, XML does not support all integration requirements at the same level as message brokers. In addition, XML can not support process integration and has an invasive nature, which increases maintenance costs.

The remaining two technologies of this category (RPCs and MOM) are not as functional as message brokers and XML. They can only support the transport layer as well as the data and custom applications integration layer. RPCs, is a mature technology that achieves reusability and facilitates real-time transactions. MOM is less mature than RPCs, it does not support real-time transactions but has higher performance. Both RPCs and MOM do not satisfy the remaining criteria.

Transaction Oriented Technologies: Both evaluation and literature support the findings that application servers are more advanced and provide more functionality than TP monitors. The latter, can be used to facilitate the transport layer as well as data, custom and packaged application integration layers. Application servers are appropriate for e-business integration as well as for packaged application integration. In addition, they can support the transport, data and the objects/components layers. Nevertheless, both application servers and TP monitors can not address integration problems at translation and process integration level. Application servers are more advanced than TP monitors and are more flexible, less complex and support maintainability and reusability. TP monitors are more mature and have high performance. Both application servers and TP monitors need to collaborate with other technologies (e.g. message brokers) to achieve enterprise integration.

Distributed Object Technologies (DOT): Clearly, DOT technologies such as CORBA, DCOM, COM and EJB can be used to provide objects and component integration. Although all DOT technologies support object integration, developers should be thoughtful when choosing a DOT technology, as DOT has low or medium portability. As a result, DCOM and COM are best for Windows based environments, EJB is preferable for Java application environments, and CORBA is the best for back-end systems integration.

Apart from the process layer, DOT technologies facilitate the integration of data and objects. DOT technologies satisfy transportation layer but only CORBA supports the translation layer. No conclusions are derived regarding the *process, translation and process automation* evaluation criteria, due to the lack of information. DOT are not mature enough, they have an invasive nature, medium performance and portability. They lead to objects/components reusability and can be used for the development of flexible and maintainable solutions. DCOM/COM and EJB satisfy real-time criterion.

Interface Oriented Technologies: Screen wrappers, APIs and adapters address the transformation of data (or objects/components) from the source application to the target application format. However, they can not support the transportation or the process automation layers. At application level, screen wrappers are the best solution for custom applications integration since, they extract and input data/objects from custom systems. APIs, are often used for the integration of packaged applications. Adapters can also facilitate the integration of packaged applications as well as the incorporation of e-business solutions. In many cases, APIs and adapters collaborate to address the integration of packaged solutions. For instance, API extracts data from an ERP system where an adapter translates and formats these data to target application. Although interface oriented technologies address data and objects integration they fail to support the incorporation of processes. However, process integration can be achieved when adapters are combined with message brokers (Linthicum, 2000).

3.5 Interpretation of Evaluation of Integration Technologies

This section discusses the findings of the evaluation of integration technologies that derive from Table 3.4. The interpretation of evaluation is presented below and it is based on: (a) application elements; (b) integration layers and, (c) classification of system types being integrated. The main findings of the evaluation are presented below:

3.5.1 Interpretation of Evaluation based on Integration Elements

The main evaluation findings for *application elements* are presented in Table 3.5. This Table classifies EAI technologies that can be used to support: (a) data; (b) objects and, (c) business processes integration. The Table 3.5 summarises only those technologies that received a *medium* or *high* value during evaluation (see Table 3.4). In presenting only these technologies, the more common used combinations of EAI technologies that are being used by developers to integrate application elements are highlighted. The main findings are discussed in the following paragraphs.

Category of Integration Technologies		Application Elements		
		Data	Objects	Process
	Database Oriented Middleware	▪ ODBC ▪ JDBC		
	Message Oriented Technologies	▪ RPC ▪ MOM ▪ Message broker ▪ XML	▪ XML	▪ Message broker
	Transaction Based Technologies	▪ TP Monitor ▪ Application server	▪ Application server	
	Distributed Object Technologies	▪ CORBA ▪ COM/DCOM ▪ EJB	▪ CORBA ▪ COM/DCOM ▪ EJB	
	Interface Oriented Technologies	▪ Screen wrappers ▪ APIs ▪ Adapters	▪ Screen wrappers ▪ APIs ▪ Adapters	

Table 3.5: Evaluation based on Application Elements

Data Integration: Although, all integration technologies facilitate the integration of data, there are significant differences among them and the solution that each one provides. For instance, database oriented technologies (ODBC and JDBC) support the extraction and insertion of data from source database to target. In limited cases ODBC, is used to integrate applications data that runs in mainframe environments. Also ODBC supports non-mainframe based applications, where JDBC often focuses on windows and java based systems. In cases of custom applications and packaged solutions screen wrappers and APIs respectively can be used to extract the data from these systems. As ODBC, JDBC, screen wrappers and APIs can not move data from one application to the other they have to collaborate with other technologies thus, providing this functionality. Thus, message oriented technologies (like MOM), transaction oriented and DOT transmit the data from one application to another. In addition, database and interface oriented technologies as well as message brokers, XML and CORBA can be used for the transformation of data to the right format.

Objects Integration: Application servers, XML, DOT and interface oriented technologies can ease the integration of objects. All these technologies support different types of solutions. Screen wrappers capture data from custom applications and transform them into objects. Screen wrappers are capable to support mainframe based applications and thus, facilitating the integration of custom systems. APIs are used to gain access to the objects/components of package solutions, so as to facilitate incorporation. Adapters act as intermediaries among package applications and other systems to translate and format the objects. XML transforms objects into messages and transfers them to a target application. Application servers are also used to transfer objects. DCOM/COM, EJB and CORBA support the integration of objects in a windows environments, Java applications and back-end solutions respectively.

Process Integration: Message brokers are the only technology that supports this layer. Message brokers translate data, share the application logic and rules, and support the integration of processes. Based the on interpretation of data as well as on the business rules and logic, they can distribute data to the target application at the right time. In addition, message brokers can activate and route the right functions.

3.5.2. Interpretation of Evaluation based on Integration Layers

In this section, the interpretation of evaluation focuses on the integration layers. Like the analysis followed in the previous section, the Table 3.6 summarises those technologies that have received medium and high values during the evaluation. The next paragraphs discuss the main evaluation findings.

Transportation layer: As it is presented in Table 3.6 message oriented, transaction based and DOT technologies support the transportation layer. It can be concluded that message brokers are a preferable technology at this level, as it conforms to the majority of evaluation criteria (see Table 3.4). However, message brokers can not sufficiently support real-time and objects integration. In these cases, XML, application servers and DOT technologies can be employed to address these integration issues.

		Integration Layers		
		Transportation	**Translation**	**Process Automation**
Category of Integration Technologies	**Database Oriented Middleware**		▪ ODBC ▪ JDBC	
	Message Oriented Technologies	▪ RPC ▪ MOM ▪ Message broker ▪ XML	▪ Message broker ▪ XML	▪ Message broker
	Transaction Based Technologies	▪ TP Monitor ▪ Application server		
	Distributed Object Technologies	▪ CORBA ▪ COM/DCOM ▪ EJB	▪ CORBA	
	Interface Oriented Technologies		▪ Screen wrappers ▪ APIs ▪ Adapters	

Table 3.6: Evaluation based on Integration Layers

Translation layer: On a translation layer, ODBC and JDBC can be used to transform data from one database to another. Screen wrappers map and translate data from one screen to a data file, or an object. APIs and CORBA can also translate objects or data from one application to another. XML transforms and translates information from one application to an XML message and vis-a-versa. Message brokers and adapters also support the translation layer. As mentioned above, the combination of message brokers with adapters provide a functional solution, as it supports non-invasive solutions, which are more flexible and maintainable.

Process Automation layer: On process automation layer, message brokers are the only solution available since the rest of the integration technologies do not support the process automation layer.

3.5.3. Interpretation of Evaluation based on the Classifications of System Types

The following sections evaluate the proposed framework, and explain the permutations of integration technologies when mapped against classifications of system types.

3.5.3.1 Custom to Custom Integration

The integration of custom to custom applications can be achieved either at a database or interface level. As reported in section 3.3.5, user interfaces are the most common points of access for custom systems. As a result, screen-wrapping tools can be adopted to extract data, business rules and objects from custom applications. After extracting the data, message oriented technologies should take place to transfer data from source application to the central integration infrastructure (e.g. message broker). Based on the evaluation results in Table 3.4, message brokers can transfer more effectively the data from custom applications to a central integration infrastructure. The reason for this is that XML is designed for Internet based applications and thus, does not support custom systems integration. In addition, MOM technology is not as flexible and manageable as message brokers, as the former has an invasive nature and therefore, leads to non-flexible solutions. MOM has many drawbacks when compared to message brokers (e.g. MOM has low portability, scalability and does not support reusability of data and processes). When the *screens as objects method* is followed, CORBA also could be used. COM/DCOM are mainly designed for windows based environments and therefore, can not support custom applications integration. EJB focuses more on Java oriented platforms and thus, it can not also be used.

Category of Integration Technologies	Classification of System Types						
	Custom to Custom	Custom to Packaged	Custom to e-business	Packaged to Packaged	Packaged to e-business	E-business to e-business	Custom to Packaged to e-business
Database Oriented Middleware	▪ ODBC	▪ ODBC ▪ JDBC	▪ ODBC ▪ JDBC	▪ ODBC ▪ JDBC	▪ ODBC ▪ JDBC	▪ ODBC ▪ JDBC	▪ ODBC ▪ JDBC
Message Oriented Technologies	▪ MOM ▪ Message broker	▪ MOM ▪ Message broker	▪ MOM ▪ Message broker ▪ XML	▪ MOM ▪ Message broker ▪ XML	▪ MOM ▪ Message broker ▪ XML	▪ MOM ▪ Message broker ▪ XML	▪ MOM ▪ Message broker ▪ XML
Transaction Based Technologies	▪ TP Monitor	▪ Application server	▪ Application server	▪ Application server	▪ Application server	▪ Application server	▪ Application server
Distributed Object Technologies	▪ CORBA	▪ CORBA ▪ COM/DCOM	▪ CORBA ▪ COM/DCOM ▪ EJB	▪ CORBA ▪ COM/DCOM ▪ EJB	▪ CORBA ▪ COM/DCOM ▪ EJB	▪ CORBA ▪ COM/DCOM ▪ EJB	▪ CORBA ▪ COM/DCOM ▪ EJB
Interface Oriented Technologies	▪ Screen wrappers	▪ Screen wrappers ▪ APIs ▪ Adapters	▪ Screen wrappers ▪ APIs ▪ Adapters	▪ APIs ▪ Adapters	▪ APIs ▪ Adapters	▪ APIs ▪ Adapters	▪ Screen wrappers ▪ APIs ▪ Adapters

Table 3.7: Evaluation based on System Types

In those instances where custom systems consist of databases that can be accessed through database connectivity drivers, ODBC drivers are the more appropriate. The reason for this is that ODBC is designed to access databases on disparate platforms and operating systems, where JDBC supports more Java oriented databases. After extracting the data from custom applications' databases, they are sent to a central integration infrastructure using message technologies (as described above). Once the central integration infrastructure receives the data from the source application, it transforms it into target applications format and distributes them according to business rules and logic. The central integration infrastructure (e.g. hub and spoke) is implemented using a message broker in most of the cases.

3.5.3.2. Custom to e-business Integration

The integration of custom to e-business applications can be accomplished at a data, objects and/or interface level. A significant part of integration in e-business systems deals with the exchange of data among two or more applications. XML simplifies integration by allowing systems to exchange information without having knowledge about the participating applications. Source application simply translates its data to XML, and then sends them to the target. The receiver extracts the information from the XML message and transforms them into its structure. In addition, XML is flexible and offers bindings for object models (e.g. CORBA, COM) and other languages, as well as support for the communication between components. This is important for e-business applications, as many of them have been implemented using a component based approach and thus, require compatible integration technologies (Morgenthal, 1999). In addition, XML can be used for sharing data among Java and non-Java environments and also, can collaborate with Java to develop dynamic applications by binding Java components based on XML document type. This allows companies to build more flexible integrated applications.

Although XML can support the integration requirements of e-business applications, it is not recommended for custom application integration. The reason for this is that XML can not efficiently support non Internet based applications. Therefore, XML should be combined with other technologies to facilitate custom to e-business integration. Since custom applications can not interpret XML messages, there is a need to send them to a central integration infrastructure. The latter is implemented by a message broker that transforms XML messages into custom applications format. In doing so, message brokers may use adapters or rely on their own technology. Based on business rules and logic, message brokers distribute the messages to custom applications. Likewise to custom-to-custom integration, data are extracted from messages and inputted in custom systems using screen wrappers. Also, ODBC might be used if access to custom applications databases is possible. Furthermore, when objects are transferred from e-business applications to custom systems, CORBA might be used along with message brokers, as the latter do not fully support objects incorporation.

3.5.3.3. Custom to Packaged to e-business Integration

In a common used inter-organisational application integration scenario (e.g. e-supply chain integration), custom systems are incorporated with packaged and e-business solutions. Having already analysed appropriate permutations of technologies that are used to incorporate custom and e-business solutions with the central integration infrastructure, this section explains how packaged applications are pieced together with the rest types of systems.

Many of packaged solutions (e.g. ERP systems) use APIs to facilitate applications communication and integration. APIs provide access to packaged systems data, objects, services and processes and can achieve real time integration. However, APIs can not transfer data or objects from packaged applications to the central integration infrastructure. As a result, DOT technologies like CORBA and COM/DCOM can be used to support this limitation of APIs. COM/DCOM is more appropriate in windows based packaged systems whereas CORBA is a non-windows based environments. Apart from CORBA and COM/DCOM message brokers can transfer information from packaged systems to the central integrated infrastructure. In addition, a message broker is required to co-ordinate and synchronise the whole integration tasks.

3.6. A Novel Model for the Adoption of Application Integration

This section, attempts to contribute in the area of EAI adoption by proposing a novel model for the introduction of application integration. Due to EAI literature limitations, the author focuses on other relevant models and research that supports the adoption of integration technologies. As a result, a number of models that were proposed for the adoption of similar technologies like Electronic Data (EDI) are taken into consideration. Such models include those proposed by Iacovou *et al.*(1995), Van Heck and Ribbers (1999), Zinner (1999) Chwelos *et al.* (2001) and Ling (2001).

The explanation for taking into consideration models that focus on EDI adoption is that: (a) EDI is considered as an integration technology, (b) it focuses on cross-enterprises applications and e-business integration and, (c) EAI follows the same concepts as EDI (e.g. extracts data from source application, translates and reformats them, transfers them to target application etc). However, enterprise application integration performs these tasks in a more flexible and advanced way (e.g. by using non-invasive technologies, achieving process integration).

Based on comprehensive literature review on application integration, as well as the EDI adoption, eight factors were identified as the main reasons that could explain the EAI adoption. Two of these factors, are introduced by the author and are based on the work presented in earlier stages of this chapter (see sections 3.1, 3.2, 3.3 and 3.4). These two factors deal with: (a) the IT infrastructure of an organisation and, (b) the existence of an evaluation framework for the assessment of integration technologies. The remaining six factors are derived from the review of other relevant models and include only these factors that are considered by the author as important for EAI adoption. Thus, the author suggests that factors like: (a) benefits; (b) barriers; (c) costs; (d) external pressure; (e) support and, (f) IT sophistication influence organisations when taking their decisions for EAI adoption. The factors of the proposed conceptual model are analysed below:

- **Benefits.** Similar to the model proposed by Iacovou *et al.*(1995) benefits refer to the level of recognition of a relative advantage that application integration can provide the organisation. Published case studies on application integration such as those reported by Edwards and Newing (2000), Anonymous (2001) and Puschmann and Alt (2001) support that organisations assess all types of benefits (e.g. managerial, technical) that EAI offers before proceeding to the introduction of this new technology. Thus, this factor differs from previous models like Chwelos *et al.*(2001) which consider only technical benefits as a factor that influences the adoption of an integrated technology. In the proposed model, *Benefits* are extended to cover: (a) operational (e.g. reduces costs); (b) managerial (e.g. increases performance); (c) technical (e.g. results in flexible infrastructures); (d) strategic (e.g. achieves customer satisfaction) and, (e) organisational costs (e.g. allow organisations to do business more effectively).

- **Barriers.** The introduction of ERP systems presents a similar case to application integration. Like ERP systems, application integration: (a) is promising to integrate IT infrastructure; (b) introduces changes to the organisation structure and the way of doing business; (c) influences the employees tasks as well as inter-organisational relationships; (d) it costs a lot of money and, (e) is more likely adopted by big organisations. Since there is a lot of failure on ERP adoption, organisations tend to estimate the possible impact of the adoption of application integration before proceeding to its adoption. Barriers are also reported by Van Heck and Ribbers (1999), Zinner (1999) Chwelos *et al.* (2001) and Ling (2001) as a factor that influences the adoption of EDI technology. Thus, the author suggests that the barriers of EAI is a factor that influences its adoption.

- **Costs.** Organisations are often reluctant to proceed to a new investment before justifying its cost and expected benefits, with Irani (1998) and Irani *et al.* (1997; 1998) exploring more of this issue. Thus, many organisations conduct a cost benefit analysis before taking an important decision. In this context, the author proposes that the costs associated with the introduction of application integration are considered as an influential factor for EAI adoption. Organisations may abandon their plans for EAI adoption, in case the cost of adoption and its negative impact (barriers) on organisation is bigger than the expected benefits. Financial resources as a factor are also reflected in various EDI models like Iacovou *et al.* (1995), Van Heck and Ribbers (1999) and Chwelos *et al.* (2001).

- **External Pressure.** Increased competition pushes organisations to search for new ways to increase their productivity and achieve competitive advantage. In addition, there is a need to easily adapt to the changing business environment. For that reason, organisations turn to EAI to achieve competitive advantages (Kalakota and Robinson, 2001; Linthicum, 2000). Another form of external pressure is the pressure from trading partners'. Customers' and suppliers' often demand closer collaboration. Therefore, enterprises are looking for new practices to better co-ordination cross-enterprise business processes which is translated to a factor that influences the adoption of EAI. External pressure is also reflected in previous studies as a factor that influences the adoption decision of a technology (Davenport, 1998; Glass and Vessey, 1999; Graham and Hardaker, 2000; Makey, 1998).

- **Evaluation Framework.** As already reported in sections 2.6 and 2.8, the integration marketplace is extremely complex with a diversity of EAI products and technologies solving different types of problems. For that reason, a framework that supports organisations in decision-making for adopting EAI can be considered as a factor that influences the adoption of EAI. The aforementioned evaluation framework influences the adoption of EAI solutions since it contributes to the selection of appropriate integration technologies and tools. The reason for this is that, given an organisation's IT infrastructure, the framework suggests a combination of integration technologies that can be used to integrate this infrastructure. Therefore, proposed framework influences the adoption of application integration technologies.

- **IT Sophistication.** This factor refers to the technical expertise in the organisation. It is related with the level of understanding in addressing technical problems at an enterprise and cross enterprise level. The evaluation framework supports IT sophistication for the reasons reported in the previous paragraph.

- **IT Infrastructure.** The non-integrated nature of IT infrastructure causes numerous problems to organisations, which need to unify their information systems and fully automate their business processes. Thus, there is a need for a technology that results into a flexible, manageable and maintainable integrated IT infrastructure. The existing IT infrastructure is a factor that affects the introduction of EAI, as the needs of an IT infrastructure are often stimulus-initiate the process for adopting application integration.

- **Support.** Along similar lines with the Sumner and Holstetler (1999) approach, the author considers the support factors (e.g. vendor support, consultants support etc) as an additional factor that affects the EAI adoption. The adoption of application integration requires organisations to invest cosiderable amount of moneys on their IT infrastructure (both hardware and software). Therefore, it is essential for companies to have support from vendors and consultants. Figure 3.3 depicts the proposed framework.

The proposed model makes novel contribution at two levels. Firstly, at the conceptual level, the model incorporates factors identified in previous studies as influencing adoption of integration technologies, like EDI. The author extents these works and adapts them to the application integration area through combing factors discussed in normative literature. Thus, resulting in the development of a consistent model for the adoption of application integration. Secondly, the concepts of the proposed model can be used as a frame of references for the adoption of inter-organisational information systems. A new aspect of the proposed model is that it introduces an evaluation framework as a factor that influences the adoption of application integration. The evaluation framework clarifies much of the confusion surrounding EAI area. As reported in sections 2.6 and 2.8 this confusion forms an important barrier to EAI adoption. Thus, the evaluation framework: (a) addresses this barrier to EAI adoption; (b) increases IT sophistication and, (c) supports understanding and decision-making.

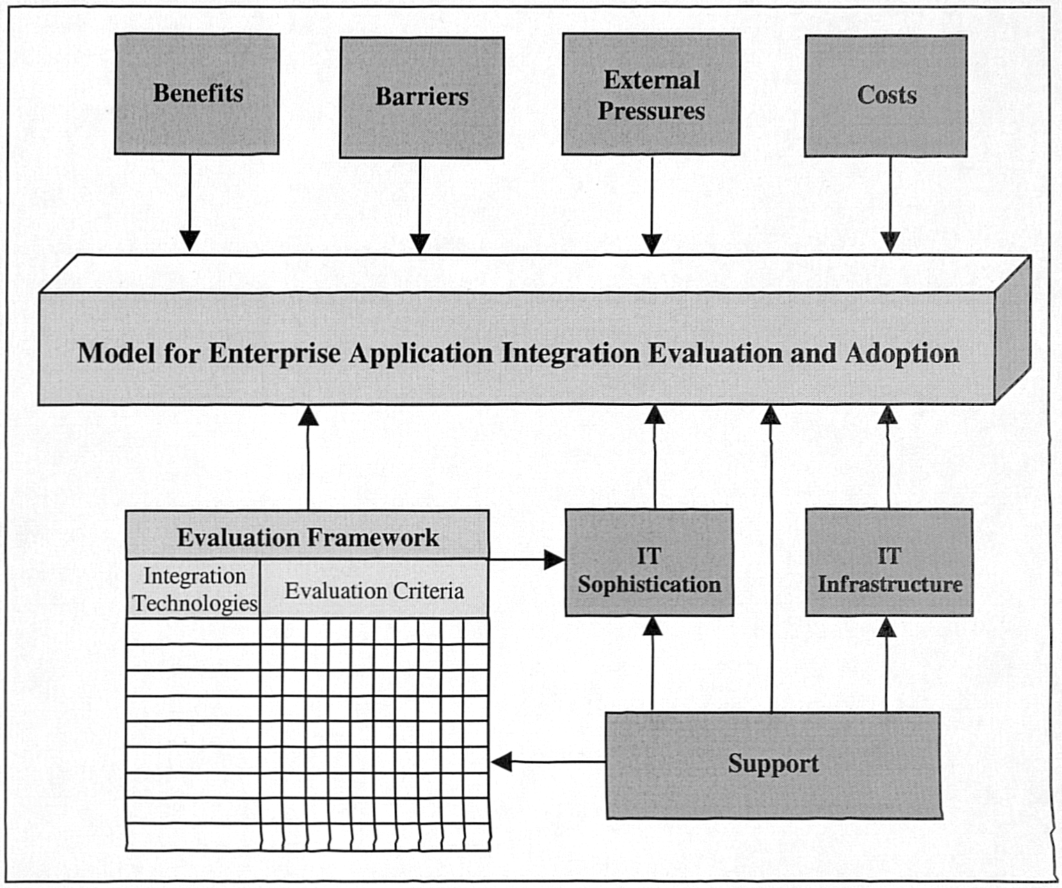

Figure 3.3: The Proposed Conceptual Model for EAI Adoption

3.7 Conclusions

The chapter starts by technically analysing enterprise application integration. Based on this analysis the following important observation has been made: *Technologies that efficiently support the integration of application elements; integration layers; classifications of system types being integrated and integration requirements of source and target applications can be used to piece applications together.*

In section 3.2 the author attempted to identify common integration requirements for the system types that are being pieced together in organisations. Much literature in software engineering (Britton and Doake, 1996; Maciaszek, 2001) supports that it is difficult to use a generalised model to identify software requirements. However, the author adopts the approach of Duke *et al.*(1999) which identifies integration requirements based on the characteristic of system types and attempts to expand this approach. Initially, common integration requirements for the classifications of system types are identified. Thereafter, based on a literature review and existing case studies on EAI, the author specifies requirements that focus on the characteristics of either EAI technologies being used to integrate systems or the overall EAI solution. These requirements can be used as evaluation criteria when assessing integration technologies.

In an attempt to clarify the technological confusion surrounding EAI, a novel evaluation framework is proposed in section 3.4 (see Table 3.4). The novelty of the framework is based on the adoption of a comprehensive set of evaluation criteria that efficiently describe application integration area. Evaluation criteria are classified into four categories (see Table 3.3) including: (a) application elements; (b) integration layers; (c) classifications of system types and, (d) common integration requirements. Application elements such as data and objects describe the elements that are used during the integration process and exchanged among applications. Integration layers refer to the stages of integration process (e.g. transportation, transformation etc) and their task is to manipulate the application elements to achieve integration. The classification of system types focuses on the range of integration in terms of system types being integrated. Integration requirements such as flexibility and portability determine those common requirements that are required when implementing integrated solutions.

Evaluation findings confirm the literature and, conclude that no single technology satisfies all evaluation criteria. Clearly, no integration technology solves all types of integration problems, as each technology was designed to address a broad category of integration issues (e.g. message integration, objects integration). Hence, a combination of integration technologies is required to achieve enterprise and cross enterprise integration. The proposed novel evaluation framework clarifies the differences among technologies, and helps integrators to select an appropriate combination of integration technologies. At a technical level, the evaluation framework presented in section 3.4 and 3.5 would appear to help developers better understand the capabilities of each technology, and highlight possible combinations of integration technologies. Strategic and business benefits may also be derived from this evaluation framework. Strategic benefits may focus on the development of an integrated infrastructure that supports enterprises to achieve competitive advantages.

Sufficient knowledge and understanding of integration technologies is important for many reasons, with organisations needing to justify their investments in IS before committing time and money to implementation. In addition, the adoption of the proposed evaluation framework leads to reusable flexible and maintainable integrated enterprise architectures, which may be achieved with minimum changes of existing applications. Such integrated solutions eliminate the maintenance effort and cost, and help enterprises in achieving competitive advantages and thus, increase their organisational performance.

This chapter then introduces a novel model for the adoption of EAI (see Figure 3.3). The model takes into consideration parameters that identified in normative literature as influencing factors. The proposed conceptual model is novel since it includes: (a) a number of consistent influential factors for EAI adoption and, (b) factors that are adapted from other relevant areas (e.g. EDI) and enriched with new factors from EAI area (e.g. the evaluation framework). The model proposes that eight factors influences the adoption of EAI namely: (a) EAI benefits; (b) EAI barriers; (c) EAI costs; (d) IT infrastructure; (e) IT sophistication; (f) external pressures; (g) support and (h) the existence of an evaluation framework that supports organisations to assess integration technologies.

Chapter 4: Research Methodology

Summary

Chapter 4 describes the research methodology of the work presented in this dissertation. This description is within the context of research methods commonly used in the area of IS. Initially, section 4.1 reviews both positivism and interpretivism epistemological stances. This review results in the justification of interpretivism as the research approach that is adopted by this dissertation. Thereafter, the author explains why qualitative research is used in this research and justifies the adoption of a case study research strategy. Then, the author presents an empirical research methodology, which acts as a framework for conducting the empirical enquiry. Finally, this methodology is transformed into a protocol, which acts as a data collection tool where data are elicited from case study companies.

4.1 Selecting an Appropriate Research Approach

Galliers (1994) and Walsham (1995a) among others report that the selection of an appropriate research approach is a major task during the research design process. The reason for this is that there are multiple methodologies to choose from with Galliers (1994) recommending methodological pluralism. In addition, the selection of an appropriate approach is not an easy task, since IS are a multi-disciplinary with many of its aspects related to natural sciences, mathematics, engineering, linguistics and behavioural sciences. Thus, there is no single framework that encompasses all the domains of knowledge needed for the study of information systems (Galliers, 1992). Orlikowski and Baroudi (1991) claim that information systems are not rooted in a single theoretical perspective, but there is a wide range of philosophical assumptions regarding the underlying nature of phenomena under investigation. Therefore, there are many research approaches and strategies that the researchers can choose.

Several philosophical approaches are available for IS research including: (a) scientific (or positivism); (b) critical; (c) interpretivism and, (d) post-positivism. These approaches rely on quite different assumptions about the nature of knowledge, and demand considerably different approaches to research, with Irani *et al.*(1999) among others, having discussed their respective characteristics.

Evidence from IS literature suggests that the positivism approach has been the dominant epistemology in IS research (Galliers, 1992; Miles and Huberman, 1994; Walsham, 1995a; Yin, 1994). Orlikowski and Baroudi (1991) suggest that IS can be classified as positivist if there is evidence of formal propositions, quantifiable measures of variables, hypothesis testing, and the drawing of inferences about a phenomenon from a perspective sample to a stated population. Galliers (1992) reports that positivism assumes that observations of the phenomena under investigation can be made objectively and rigorously (e.g. by measurement). Nonetheless, positivism approach has arisen from scientific tradition and therefore, it is characterised by repeatability, reductionism and refutability.

Galliers (1992) argues that the positivism is not the only relevant approach to IS. An alternative to positivism is interpretivism, which assumes that the knowledge of reality is gained *only* through social constructions such as consciousness, shared meanings, language, documents, tools and other artefacts. Interpretivism research does not predefine dependent and independent variables but, focuses on the complexity of human sense as the situation emerges (Kaplan and Maxwell, 1994). Interpretivism research aims at the

> *"understanding of the context of the information system and the process whereby the information system influences and is influenced by the context."*
>
> Walsham (1993, pp. 4-5).

In interpretivism, researchers tend to allow concepts (constructs) to emerge from field data, rather than entering the field with pre-conceived theories (Glaser and Strauss, 1967; Miles and Huberman, 1994). Walsham (1995a) explains more this issue by reporting that whilst it is important to access existing theory in a particular subject domain, it is equally important not to assume that it represents final truth in that area.

Positivist and interpretivist have an impact on empirical research strategy, since the former dictates that the researcher takes the role of an observer, whilst the latter dictates that the researcher gains knowledge by participating in the subject of the empirical study (Irani *et al.*, 1999).

The author argues that for the purpose of this thesis, the interpretive research approach has been selected. The justification for this choice is the following:

- Literature review and analysis presented in Chapters 2 and 3 indicates that there are many political, cultural, managerial, social and technical issues related with the adoption of enterprise application integration. These factors appear to be multiple, complex and interrelated. Hence, the factors reported in Chapter 3 that influence the adoption of EAI can not be separated from its organisational, technical and cultural context. Therefore, there is a need for a research approach that will allow the author to understand the process of adopting EAI as well as all these factors that influence

EAI adoption. Interpretivism is considered by the author as more appropriate for the research reported in this dissertation for the reasons explained thus far. The justification of this decision is based on the aim of this research as stated in section 1.5.

- Positivism can not adopted in this research, since there are no research hypothesis, quantifiable measures of variables or formal propositions in the research reported. The adoption of EAI, as it is described in section 3.6, could not be viewed as one where facts and values are independent. This means that positivism can not be used in the context of this dissertation since, positivism assumes that knowledge is consist of facts that are independent.

4.2 Justifying the Use of Qualitative Research

Irani (1998) among others, reports that events that form a phenomenon are conditioned by interacting variables, such as time and culture. This indicates that no two situations are identical. Thus, it appears that quantitative research methods are inappropriate in this case, as they are unable to take account of the differences between people and the *objects of the natural sciences.* Information systems research is concerned with human beings and therefore, any methodology that uses quantitative research methods must recognise the variability that is inherent in human behaviour. The research presented in this thesis, focuses on the factors that influence the decisions of human beings (e.g. managers) when adopting and evaluating EAI solutions. As a result, the principle of scientific methods to the study of people is questioned thus, suggesting the suitability of a more qualitative approach.

It appears from the objectives of this dissertation, that the issues under investigation are confidential and subjective, with much context to the data needed. This suggests that the selected research methods must be able to take account these issues and acknowledge that many management decisions are idiosyncratic and guided by circumstances pertaining the organisation. Clearly, *rich* empirical data is required to provide more understanding regarding the EAI adoption process. The need for rich empirical data indicates that the use of qualitative research methods is appropriate, since they allow examining in depth processes.

Miles and Huberman (1994) describe qualitative research as one that is based upon words, rather than numbers. Denzin and Lincoln (1994) suggest a more appropriate definition for qualitative research, and propose that it is multi-method in focus, involving an interpretive, naturalistic approach to its subject matter. This definition implies that qualitative researchers study things in their natural environment and they understand events in terms of meanings.

Marshall and Rossman (1999) summarise some of the types of the research that qualitative research would be appropriate. These types include among others the following:

- Research that examines in depth into complexities and processes;
- Research on little-known phenomenon or innovative systems;
- Research that seeks to explore where and why policy and local knowledge and practice are at odds;
- Research that can not be carried out experimentally for practical or ethical reasons;
- Research on informal and unstructured linkages and processes in organisations;
- Research on real, as opposed to stated, organisational goals and,
- Research for which relevant variables have yet to be identified.

In addition, Benbasat *et al.* (1987) support that qualitative research approach provides many benefits such as: (a) it allows the researcher to understand the nature and complexity of the process taking place; (b) valuable insights can be gained into new topics emerging in the rapidly changing IS field (e.g. applications integration) and, (c) the researcher can study IS in a natural setting, learn about the state of the art, and generate theories from practice. These benefits allow the author to choose a qualitative approach for the research reported in this dissertation. The justification for this decision is reported at the end of this section.

However, qualitative research approach presents a number of drawbacks, which should be taken into consideration when adopting such a research approach. Disadvantages associated with qualitative approach include the fact that qualitative data has certain, rather problematic characteristics, which set it apart from quantitative data (Miles and Huberman, 1994). Qualitative data is usually predominantly textual, with a richness that can be lost when aggregation or summarisation occurs. The data can be fairly unstructured and unbounded as it concerns peoples behaviour and attempting to understand their perception of a particular situation. It is often longitudinal, to a greater or lesser extent as the observations may continue for an extended period of time. Interviews may be repeated at intervals of a few days, weeks or months. However, Lee (1991) identified the disadvantages of qualitative analysis as a lack of- controllability, deductibility, repeatability and generalisability.

Smithson and Cornford (1996) found that there are more drawbacks to qualitative research. As the research uses a small number of cases (perhaps only one case), it is difficult to generalise it to a wider range of situations. Secondly, since the data is rich and complex, it means that it is open to a number of interpretations, such that researcher bias is a constant danger. Thirdly, researchers involved in dynamic cases where the situation is changing frequently, face inherent problems in trying to make controlled observations, controlled deductions (e.g. using mathematical and statistical methods) and predictions. This causes problems to the validity and verifiability of the research.

Bearing these points in mind and due to the epistemological stance being followed in this dissertation, qualitative research was still selected to be most suitable for this research. The reasons for selecting this approach are summarised below:

- A qualitative researcher studies things in their natural settings. In doing so, the researcher attempts to understand a phenomenon (e.g. EAI adoption) in terms of the meanings that people bring to them (Denzin and Lincoln, 2000).

- A qualitative approach will allow the author to research on little-known phenomena like EAI adoption and examine in depth the adoption processes and its complexities. This will also allow the author to seek to explore where and why policy and local knowledge and practice are at odds. In addition, the research as described in Chapters 1, 2 and 3, can not be carried out experimentally.

- As explained in Chapters 2 and 3, EAI is a state of the art technology with limited literature and research. Thus, qualitative research will support the author to study EAI in its natural setting, and learn from practice. This will allow the author to understand the nature and the complexity of the EAI adoption and evaluation process.

- The issue regarding generalisations is overcomed by using Walsham's (1995b) comments in that interpretivist case studies offer four types of generalisations, thereby overcoming this particular issue. The bias that is considered to be a danger in using qualitative research approach can be overcome by data triangulation.

4.3 Selecting a Research Strategy

Having justified the use of interpretivism as an epistemological stance and, the use of qualitative research approach, this section focuses on selecting a research strategy. Galliers (1992) reports that research strategy is the means of going about one's research, taking on a particular style and utilising different research methods to collect data with. Therefore, to decide on a strategy that would dictate the way in which data is collected and analysed, different research strategies must be reviewed. Their characteristics should be identified, and a research strategy be justified in light of these study characteristics.

4.3.1 Justifying the Use of Case Studies

Klein and Myers (1999) report that case study research is accepted as a valid research strategy within the IS research community. The use of a case study represents a way to systematise observation and aims for in-depth understanding of the context of a phenomenon (Cavaye, 1996). The strategy is versatile and open to a lot of variation, and can be carried out taking a positivism or an interpretivism stance (Stake, 2000).

Yin (1994) suggests that a case study is an intensive examination of a phenomenon in its natural setting, employing multiple methods of data to gather information from one or more entities (e.g. people, groups). Data is collected via interviews, observation, questionnaires and written materials. The main characteristics of case studies reported below, as summarised by Benbasat *et al.*(1987):

- Phenomenon is examined in natural setting;
- Data is collected by multiple means;
- The complexity of the unit is studied intensively;
- The focus is on contemporary events;
- One or few entities are examined;
- No experimental controls or manipulation are involved;
- The investigator may not specify the set of variables in advance;
- The results derived depend on integrative powers of the investigation;
- Changes in the site selection and data collection methods could take place as the investigator develops new hypothesis;
- Case studies are more suitable for exploration, classification and hypothesis development stages of the knowledge building process. The investigator should have a receptive attitude towards exploration; and,
- Case research is useful in the study of *why* and *how* questions because these deal with operational links to be traced over time rather than with frequency or incidence.

Yin (1994) suggests that there are different types of case study such as exploratory, descriptive and explanatory depending on whether they are used to answer *what, how*, and *why* research questions respectively. *Based on this taxonomy, the case study followed in this dissertation can be classified as exploratory.* The reason for this is that the research focuses more on questions of *what* type (e.g. what are the factors that influence the adoption of EAI). Exploratory case studies like the one presented in this dissertation are useful for theory building as they are valuable in developing and refining concepts for further study. Roethisberger (1977) suggests that *case study research is particularly appropriate for certain types of problems such as those in which research and theory are at their early formative stages.* As stated in Chapters 2 and 3 EAI is a new area with limited research.

Thus, the use of a qualitative case study strategy is considered by the author as appropriate for studying the phenomena of EAI adoption of evaluation.

As a qualitative technique, case study strategy is used in many studies of IS. However, to the best of the author's knowledge such a strategy has not been used in the area of EAI. This is attributed to the fact that there is a lack of published scientific research regarding EAI, for the same reasons as those reported above (EAI is a new research area). Most of the EAI literature references have been published by practitioners, which also indicate lack of scientific research regarding EAI. Therefore, it is well suited to capture the knowledge of practitioners and develop theories from it, especially in areas where the researchers are lagging behind practitioners such as EAI. Thus, for all these reasons reported thus far, the author claims that case study strategy is appropriate for the research presenting in this dissertation.

4.3.1.1 Single and Multiple Case Studies

Case studies can be single or multiple and the decision to analyse one or multiple cases is a central one to case study design. A single case study would provide 'rich' primary data of the organisational context. It would enable the research to develop a full picture of the organisational idiosyncrasies and allow the author to investigate the adoption and evaluation of EAI. However, a single case may not provide sufficient data that would justify conclusions about EAI adoption and evaluation. Therefore, in the light of the characteristics of this research, a single case study will not be appropriate.

Dismissing a single case study approach suggests that multiple cases will prove more appropriate for the research proposed in this thesis. Conducting multiple cases will enable the research to examine and 'cross-check' findings. Also, the analysis of data across organisations will be possible with this strategy. Admittedly, multiple cases will not provide the 'richness' of data that a single case study can do. Multiple cases will give the research a more 'robust' investigation of cause and effect relation of the units of analysis (Herriot and Firestone, 1983), as it will be able to move the investigation form one organisational context to an other. Thus, isolating idiosyncrasies that contribute to explaining the phenomenon. The number of case studies conducted will depend on how much is known about the phenomenon, and how much information that can be uncovered for conducting additional cases (Dyer *et al.*, 1991). However, Eisenhardt (1989) suggests that a research strategy that employs multiple case studies should not conduct more than ten, and less than four cases. As such, the research in this thesis will employ the use of multiple cases studies within the limits suggested by Eisenhardt (1989). However, Gable (1994) suggests a multiple case study should include up to five companies. In the context of the research presented in this dissertation, a multiple case study strategy was adopted to study two global EAI projects and four pilot studies.

4.4 Empirical Research Methodology

Flick (1998) suggests that the use of a set of procedures that are open-ended and rigorous at the same time are important of a qualitative research design. These procedures do justice to the complexity of the social setting under investigation. Janesick (2000) proposes that a qualitative research methodology may follow three stages. Similarly, the author has developed an empirical research methodology, which is based on three stages namely: (a) research design; (b) data collection and, (c) data analysis. These stages are illustrated in Figure 4.1 and are analysed in following sub-sections.

4.4.1 Research Design

The research design is the first independent part of the empirical research methodology. The starting point is to review the literature, thus developing an understanding of the research area under investigation. From literature review, several research issues will be identified for a more focused literature review (EAI adoption). This leads to a specific research area and ultimately, identifies a research need.

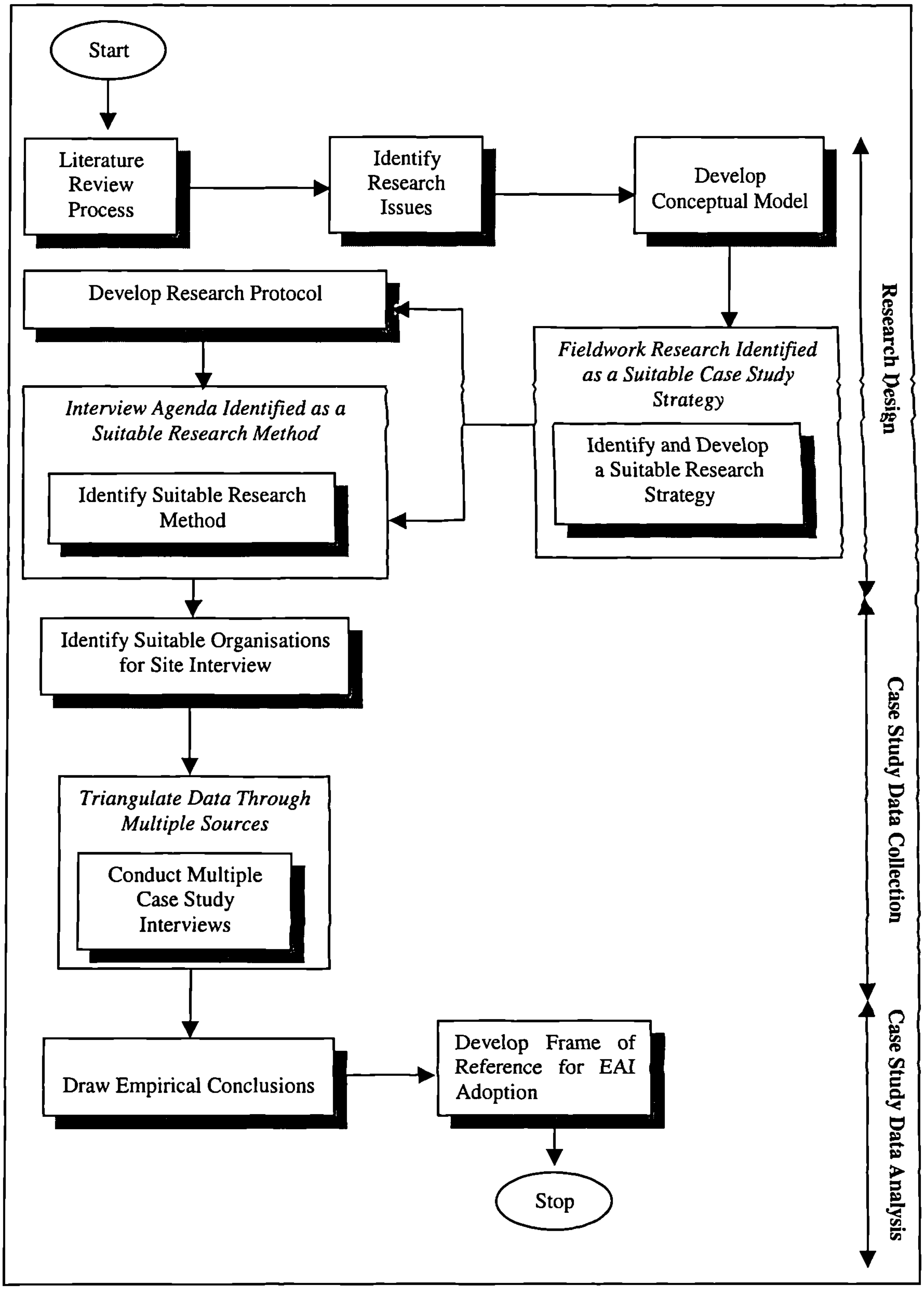

Figure 4.1 Empirical Research Methodology

Thereafter, the development of a conceptual model is conducted to represent the intended empirical research. Aspects of the model will be investigated through empirical studies. Based on the needs of the empirical study, it was decided that the research design would utilise a multi-case study strategy through the employment of qualitative research methods. The justification for selecting a multi-case study strategy is given in section 4.3.1.1. The research design was then transformed into a plan of action or protocol (see section 4.6). Research protocols are a necessary investigation tool for a number of reasons, including:

- To put the task of data gathering in a manageable format;
- To insure that targeted data is collected;
- To insure that the research follows a specific schedule;
- To track the path at which knowledge was developed; and,
- It acts as a map that others may follow to achieve similar conclusions. This is especially needed were the issues under investigation are subjective, and where the research depends on qualitative methods

Within the protocol, a qualitative research method was developed to gather data as required by the units of analysis. The method was in the form of an interview agenda (see Appendix C), which is a series of questions relating to the units of analysis, and designed to guide the researcher, during the structured interviews. In addition to the interviews, data was collected through several sources like archival documents, minuets for meetings, consultancy reports, and the website of the organisations. The use of multiple data collection methods makes the triangulation possible which provides stronger substation of theory (Eisenhardt, 1989).

4.4.2 Data Collection

Multiple data collection methods are typically employed in case studies. Using multiple methods of data collection lends greater support to the researcher's conclusions. Ideally, evidences from two or more sources will converge to support the research findings. Yin (1994) identifies several sources of evidences that can be used in case studies. These sources include: (a) documentation; (b) archival records; (c) interviews; (d) observation and, (e) physical artefacts. Table 4.1 summarises: (a) the strengths; and the weaknesses of the main sources of evidence in case studies as identified by (Yin, 1994) and, (b) provides examples of the use of these methods in this research. As presented in Table 4.1 the author has used the following methods for data collection: (a) documentation; (b) archival records; (c) interviews; (d) direct observation; (e) participant observation; and, (f) physical artefacts.

Sources of Evidence	Strengths as identified by Yin (1994)	Weaknesses as identified by Yin (1994)	Use of Sources in this Study
Documentation	▪ Stable-can be reviewed repeatedly ▪ Unobtrusive - not created as a result of the case study ▪ Exact-contains exact names, references and details of an event ▪ Broad coverage-long span of time, many events and many settings	▪ Retrievability-can be low ▪ Biased selectivity, if collection is incomplete ▪ Reporting bias-effects (unknown) bias of author ▪ Access-may be deliberately blocked	▪ Reports from the organisations under study ▪ White papers ▪ Reference material downloaded from Internet ▪ Newspaper articles
Archival Records	▪ [same as above for documentation] ▪ Precise and quantitative	▪ [same as above for documentation] ▪ accessibility due to privacy reasons	▪ Deliverables of previous project on ERP and ebusiness interconnectivity ▪ Organisational records
Interviews	▪ Targeted-focuses directly on case study topic ▪ Insightful-provides perceived causal inferences	▪ Bias due to poorly constructed questions ▪ Response bias ▪ Inaccuracies due to poor recall ▪ Reflexivity-interviewee gives what interviewer wants to hear	▪ Structured interviews ▪ Semi-structured interviews ▪ Unstructured interviews
Direct Observation	▪ Reality-covers events in real time ▪ Contextual-covers context of event	▪ Time consuming ▪ Selectivity-unless broad coverage ▪ Reflexivity-event may proceed differently because it is being observed ▪ Cost-hours needed by human observers	▪ Formal and informal meetings with interviewees for gaining further insights
Participant Observation	▪ [same as above for direct observations] ▪ Insightful into interpersonal behaviour and motives	▪ [same as above for direct observations] ▪ Bias due to investigator's manipulation of events	▪ Simple participant
Physical Artefacts	▪ Insightful into cultural features ▪ Insightful into technical operations	▪ Selectivity ▪ Availability	▪ Hardware and software and equipment

Table 4.1: Data Collection Methods – Strengths, Weaknesses and their Use in this Study

4.4.2.1 Interviews

Interviews are considered to be the main tool of the qualitative researcher for data collection (Denzin and Lincoln, 1998), and one of the frequently used data collection tools utilised for this research. Additionally, since the interpretive stance is also being followed, interviews are viewed to be the main and appropriate source from where data has been collected. According to Walsham (1995b) interviews allow the best access to the: (a) interpretations that the participants have regarding the actions and events which have or are taking place and, (b) the views and aspirations of themselves and other participants. An added benefit is that it allows researchers to step back and examine the interpretations of their fellow participants in some detail. This is an advantage that other methods may not allow.

There are various forms and types of interviews in existence. According to Denzin and Lincoln (1998) there are three major types of interviews namely: (a) structured; (b) semi structured and, (c) unstructured. Interviews can also be undertaken in various forms like personal interviews, face-to-face group interviewing, telephone surveys etc. The duration of an interview is also not specific, as it could last as a five minutes conversation on the telephone, or it could take place over lengthy, multiple sessions (Frey and Fontana, 1991).

In the context of this research, interviews constituted the main data source in the cases. Three people in each organisation under investigation were interviewed using structured (and semi-structured or unstructured) interviews. Structured interviews were based on the interview agenda presented in Appendix C. Using the interview agenda, the interviewees replied in specific questions regarding EAI adoption and evaluation. Semi-structure interviews took place without the use of an interview agenda. Using this type of interviews the author attempted to clarify some issues that derived from structured interviews. In the majority of cases, structure or semi-structured interviews took place at interviewees' office. Unstructured interviews dealt with discussions that the author had with interviewees but without using a structured or semi-structured type of interview. The author had unstructured interviews during lunches, coffee breaks etc. Using unstructured interviews some important data regarding the case studies were collected (e.g. inside information regarding resistance to change, politics issues).

In both case studies, interviewees selected for structured interviews included: (a) a project manager; (b) an integrator and, (c) a consultant (internal or external) all of whom have been directly involved in the EAI projects. Such stakeholders had an important role during the decision making process for EAI adoption and evaluation as well during the implementation of the EAI projects. Therefore, it was considered important to select a cross section of roles in the EAI projects to obtain the views of stakeholders at different levels in the organisations. This supports better understanding of the phenomenon of EAI adoption and evaluation. All of the interviews were tape recorded and transcripts prepared as soon as possible after each individual interview. Tape recording supported the author in collecting accurate data and interpreting them without time pressures. The availability of interviewees was a problem during the case studies, since they were too busy and therefore, there was limited time for interviews. Taking notes during the interviews simply reduces the time of interviews since notes taking requires more time. Thus, the author considered tape recording, as a more effective way of conducting interviews.

Apart from the structured interviews, during the research there were opportunities to obtain views of other users, managers, developers via informal meetings and unstructured interviews. Table 4.2 summarises the design of data collection through interviews.

The interview agenda is summarised in Appendix C and it focuses on collecting data from the following areas:

- **General Background:** This section attempts to collect general information regarding the organisation under study. Such data include among others: (a) the number of employees in the organisation; (b) the key business of the organisation; (c) the number of subsidiaries and the (d) the nature of the organisation (e.g. multinational).

- **Technical Information:** The purpose of this section is mainly to collect data: (a) regarding the existing IT infrastructure and the problems that this infrastructure causes to the organisation; (b) to identify the EAI solution; (c) to identify the types of systems that are integrated (and examine if these types of systems follow the classification proposed in section 2.4); (d) to identify the importance of the criteria proposed in section 3.4 for the evaluation of integration technologies and, (e) to search for technical factors that influence the adoption of EAI.

- **Business Information:** The last section of the interview agenda aims at collecting the data related to other influential factors for EAI adoption. Data collected through this section (business information) deal with factors like: (a) costs; (b) benefits and, (c) barriers.

Such an interview agenda covers all the important issues that were identified in Chapters 2 and, 3 and dealt with the factors that influence EAI adoption and the development of an evaluation *framework for integration technologies.*

Organisation	Type of Interview	Respondent Position in EAI project	Type of Interview
Case Study 1	▪ Face-to-Face ▪ Telephone	Project Manager	▪ Structured ▪ Semi-structured ▪ Unstructured
Case Study 1	▪ Face-to-Face ▪ Email questions ▪ Telephone	Integrator	▪ Structured ▪ Semi-structured ▪ Unstructured
Case Study 1	▪ Face-to-Face ▪ Email questions ▪ Telephone	Internal Consultant	▪ Structured ▪ Semi-structured ▪ Unstructured
Case Study 1	▪ Face-to-Face ▪ Email questions	Developer	▪ Semi-structured ▪ Unstructured
Case Study 1	▪ Face-to-Face	Internal Consultant	▪ Unstructured
Case Study 1	▪ Face-to-Face	Internal Consultant	▪ Unstructured
Case Study 2	▪ Face-to-Face ▪ Email questions ▪ Telephone	Project Manager	▪ Structured ▪ Semi-structured ▪ Unstructured
Case Study 2	▪ Face-to-Face ▪ Email questions ▪ Telephone	Integrator	▪ Structured ▪ Semi-structured ▪ Unstructured
Case Study 2	▪ Face-to-Face ▪ Email questions ▪ Telephone	External Consultant	▪ Structured ▪ Semi-structured ▪ Unstructured
Case Study 2	▪ Face-to-Face	Developer	▪ Unstructured
Case Study 2	▪ Face-to-Face ▪ Email questions	IT Manager	▪ Unstructured

Table 4.2: Data Collection Design via Interviews

4.4.3 Data Analysis

Data analysis is the third part of the empirical research methodology presented in Figure 4.1. Empirical data derived from case studies were triangulated (see section 4.5) and then analysed to draw empirical conclusions. A difficulty in the use of qualitative data is that the methods of analysis are often not well formulated (Miles and Huberman, 1994). The process of qualitative data analysis takes many forms but, is fundamentally non-mathematical in nature. During this study, data analysis has involved examining the meaning of people's words and actions. Similarly to other studies (Galal, 1996; Irani, 1998; Ramanath, 2000), the research findings of this study are inductively derived from empirical data. Empirical evidences were then used to draw conclusions and resulted in the formulation of a frame of references for EAI adoption and evaluation.

4.5 Data Triangulation

Another important issue that concerns interpretive researchers is the validity and reliability of research findings. The term that is usually related with those issues is that of triangulation as means of validating the results (Denzin, 1978). Denzin (1978) suggested that there are four types of triangulation namely: (a) data; (b) investigator; (c) theory and, (d) methodological. Janesick (2000) adding a fifth type called interdisciplinary triangulation.

Data triangulation means the use of variety of data sources in a study (Denzin, 1978). The second type of triangulation is the investigator triangulation, which is the use of several different researchers or evaluators (Janesick, 2000). According to Denzin (1978) *Theory triangulation* refers to the use of multiple theoretical perspectives to interpret a single set of data. *Methodological triangulation* means the use of multiple methods to study a single problem. Finally, *Interdisciplinary triangulation* is related with the investigation of issues related with more that one disciplines (Janesick, 2000).

From these definitions, it can be concluded that data, methodological and interdisciplinary triangulation are being employed in this research and these results are summarised and illustrated in Table 4.3.

Organisation	Type of Triangulation Applied	Sources
Case Study 1	Data	▪ Reports, ▪ White papers ▪ Interviews ▪ Deliverables ▪ Organisational records ▪ Observations
	Methodological	▪ Documentation ▪ Archival records ▪ Interviews ▪ Observations ▪ Physical artefacts
	Interdisciplinary	▪ Information Systems ▪ Management ▪ Culture
Case Study 2	Data	▪ Reports, ▪ White papers ▪ Newspaper articles ▪ Internet resources ▪ Interviews ▪ Organisational records ▪ Observations
	Methodological	▪ Documentation ▪ Archival records ▪ Interviews ▪ Observations ▪ Physical artefacts
	Interdisciplinary	▪ Information Systems ▪ Management ▪ Culture

Table 4.3: Types of Triangulation Used in the Research

In the initial interview (whether telephone or face to face), questions relating to the role of individuals, backgrounds of the organisation and general facts about the project were asked. These questions were open-ended, as the researcher wanted to obtain as much information as possible and not limiting the respondent in any way. In some cases this has led interviewees to report issues that had not taken into consideration by the author (e.g. support factors) during the designing of interview-agenda.

4.6 Case Study Protocol: An Operational Action Plan

A case study protocol was described by Yin (1994) as a tool that would operationalise the research, acting as an action plan, and setting rules and regulations by which data would be gathered. The protocol acts as a data collection tool, where data are derived from case studies. Such a protocol is necessary to increase the consistency and focus of the data gathering process (Remenyi, 1991).

The necessity of having a case study protocol was discussed by several researchers, including Irani *et al* (1999). The argument was that in situations were the empirical inquiry was subjective, and seem to depend on irregular data gathering tools, then, a scientific map of the research must be developed so that other researchers can trace the path of data collected, and ultimately, knowledge created. As such, the case study protocol represents an official document that an investigator uses to schedule data gathering dates, to specify the means by which it will be gathered, and to detail the objectives and procedures of the analysis. Yin (1994) suggests that case studies may have questions at five levels, as presented in Table 4.4. He added that a case study protocol will outline: (a) the case study overview; (b) fieldwork research procedures; (c) questions addressed by the research, and, (d) the research output format. As such this thesis will adopt the outline suggested by Yin, and this chapter will address level 1, 2 and 3 questions, with other parts of the thesis addressing the remaining levels.

Question Level	Research Question	Section Reference
Level 1	Questions asked of specific interviewees	4.6.2.1 / Appendix C
Level 2	Questions asked of an individual case study	4.6.1 / 4.6.2 / 4.6.3
Level 3	Questions asked across multiple case enquires	4.6.3
Level 4	Questions asked of entire study	1.5 / 7.1
Level 5	Questions about the recommendations and conclusions beyond the scope of the study	7.4

Table 4.4: Questioning Levels in a Multiple Case Enquiry Source: Yin (1994)

4.6.1. Case Study Overview

The author suggests that it is not the intention of this research to offer prescriptive guidelines to EAI adoption and evaluation but rather, describe case study perspectives that allow others to relate their experiences to those reported. Hence, this dissertation offers a broader understanding of the phenomenon of enterprise application integration adoption and evaluation.

In this section of the case study protocol, the issues under investigation are detailed, to assist the researcher in focusing on the main questions that need to be studied. These are factors that the author needs to focus on, to generate data that is required to investigate the adoption and evaluation of EAI. The consideration of these issues are crucial, to retain focus during the interviews. These issues are the following:

- To identify the EAI adoption decision-making process used by the case study organisations;
- To identify those human, organisational and technical factors associated with the adoption and evaluation of EAI, and identify their suitability for inclusion in a conceptual model for EAI adoption and evaluation;
- To identify those stakeholders associated with the adoption of EAI and justify their contribution to the decision-making process for EAI adoption and evaluation;
- To identify the portfolio of benefits, barriers and costs considered during the introduction of enterprise application integration; and,
- To identify these evaluation criteria considered during the evaluation of integration technologies and EAI solutions.

4.6.2. Fieldwork Research Procedures

As reported in section 4.3.1 the nature of cases studies is related to the examination of a phenomenon in its natural (real-life) setting. This means that the researcher should take into consideration and cope with 'real world' events such as respondents dropping out, documents not being available etc. Obviously appointments with interviewees will be scheduled, and documents can be requested ahead of time, but they will never be guaranteed. Furthermore, interruptions during the interview are expected, and documents may not be available, but that should not stop the investigator from collecting data. Thus, a fieldwork procedure must be designed to cope with such events. This section of the protocol presents those procedures that will be employed during the multi-cases study investigation. These procedures include the following:

- Specify who need to be interviewed: IT manager in all cases needs to be interviewed, as well as consultants, and developers. Since EAI is an emerging technology, there is often lack of knowledge in organisations. Thus, similarly to the introduction of other technologies that focus on integration (e.g. EDI, ERP) organisations seek support from consultants. In many cases, consultants influence the decision making process for the adoption of a technology (Chung and Snyder, 2000; Oliver and Romm, 2000). Also, the author considers IT managers and developers as stakeholders that need to be interviewed due to their position and role during the adoption of a new technology.

- Identify appropriate data gathering research methods and establish line of inquiry. Interview agenda (see Appendix C) was developed and used to collect rich primary data through structured interviews. The agenda enabled the 'steering' of the interview process. Interviews were tape recorded, and transcribed on a later date. Additional data to support findings was obtained from archived documents, meeting minutes, reports, the website of the organisation and other sources as presented in Table 4.1.

- Develop data collection agenda that takes into account contingencies: in the case on failure by the interviewee to keep the appointment, predefined employees should be on 'stand-by'. In reality, this has proved difficult to operationalise and in few instances the author had to wait or interviews did not appear.

- Develop an interview timetable: dates and times were set to accommodate the interviewee needs. All interviewees were asked to set aside at least one hour for the meeting. In many cases interviews took much longer (especially those took place abroad).

- Identify and discuss supplementary framework procedures: to insure full disclosure of information, each interviewee was given a confidentiality agreement. The agreement also applied to the organisation as a whole. Essentially, the organisation agreed for the information to be published if its specific identity was not disclosed.

Thus, the author refers to the organisations under investigation without using their names but by reporting them as OILCORP and AUTOCORP.

Conducting interviews requires the skill of distinguishing between what is relevant, and what is added by interviewees. It also requires the ability to make interviewees discuss issues that may be controversial and confidential. Therefore, the first step is to gain the confidence of the interviewee by establishing the presence of the confidentially agreement, in which all information disclosed will be represented without indication to the identity of the provider. Interviewees were asked to start by describing their function in the organisation. This was done to put the interviewee at ease, and in the mode for the interview. Tape recording was required to capture the data, therefore it was requested form each interviewee. Once the interviewee was comfortable with the process and the presence of the tape recorder, the interview agenda was used to guide the structure interviews. Upon discussing a certain point, the interviewee was not interrupted since it often leads to the disclosure of relevant data. Obviously, if the discussion strayed beyond the scope of the empirical inquiry, then 'steering' was applied.

To verify the accuracy of the data, at least three staff members were interviewed in each case with the same line of questing. In addition, when applicable, organisational documents were produced to support the claims of the interviewee. Thus achieving what researchers label as data triangulation (Yin, 1994).

4.6.3. Questions Addressed by the Research

In maintaining focus on the task of data collection, a set of questions was developed. These questions are set for the researcher, and not for the interviewees and act as a reminder for the researcher, concerning the data. This data is essential to be collected to investigate EAI adoption and evaluation. Interviewees are not exposed to these questions, but were used for consultation before and during the interviews to maintain some form of structure to the interview. Essentially, the main purpose of the protocol questions are to keep the interviewers (author) focus during the data collection process. The author had an opportunity to review key questions that the interview should address. For that reason four questions were developed to be asked of the interviewee and represent part of the questions level 2 in Table 4.4. Table 4.5 summarises these questions.

Question Number	Question
1	What are the factors used by the case organisations that influence the decision making process for EAI adoption and evaluation?
2	What are the human, organisational and technical factors that associated with EAI adoption?
3	What are the benefits, barriers and costs associated with EAI adoption?
4	What are the evaluation criteria used by the case organisations during the evaluation of integration technologies and EAI solutions?

Table 4.5: Questions Addressed by the Empirical Inquiry

4.6.4. The Research Output Format

Chapter 5 presents the empirical data analysis, and the format at which the output of the empirical inquiry will take. The consideration of the format that the research output should take proved useful, as large amounts of data would be gathered during each case study visit. The author addressed issues associated with large amounts of data likely to be generated, through aligning each question within the interview agenda. This approach contributed to the quality of the research output, as it focused on the development of an effective interview agenda for the investigation of the factors that influence the adoption and evaluation of EAI.

4.7 Conclusions

The aim of this chapter is to propose a rational for the use of an appropriate research methodology for this dissertation. This chapter examines the research methodology to be applied within this dissertation. This does not only provide the research process with a well-developed framework but, provides an understanding in the broadest possible terms.

A discussion of the epistemological stances and their suitability was initially provided. In doing so, the author has justified the use of an interpretivism stance for the research presented in this thesis. The reason for this decision is based on the aim of this research as described in section 1.5 and deals with the development of a frame of references that will support decision making during the adoption and evaluation of EAI. Thereafter, quantitative and qualitative research approaches are discussed. The author suggests that in the context of this research qualitative approach is more appropriate for the reasons explained in section 4.2. Such reasons include that qualitative approach can be used to: (a) investigate little-known phenomena like EAI adoption; (b) examine in depth complex processes (EAI adoption); (c) examine the phenomenon in its natural setting and, (d) learn from practice.

In section 4.3, the types of research strategies that are available and reasons for selecting particular ones were provided. Thus, the use of case study strategy in this research was justified and explained in section 4.3.1. Furthermore, multiple case studies are used within this research to explore and understand the adoption and evaluation of EAI. In addition, the use of research methods was outlined and discussed and arguments for the suitability of particular methods were provided. Thus, various methods for data collection are used by the author during this research including among others: (a) interviews; (b) documentation; (c) observation; (d) archival records and, physical artefacts.

Then, sections 4.4 and 4.5 reported the: (a) empirical research methodology followed in this research and, (b) data triangulation respectively. Thereafter, section 4.6 presents the case study protocol for this research. This protocol can be used as an important tool that acts as an operationalised action plan for the empirical enquiry. Based on this protocol the author will use case study perspectives to allow others to relate their experience to the outcome of this research. Thus, thc work presented in this dissertation will provide a broader understanding of the phenomenon of EAI adoption and evaluation.

5

Chapter 5: Case Studies and Preliminary Research Findings

Summary

This chapter presents and analyses empirical data that is used to test the proposed model for the adoption and evaluation of EAI. However, the analysis of the empirical data should not be seen as a comparison among cases. Instead, this chapter offers an empirical analysis of different case study perspectives that describe human and organisational behaviour and perceptions during the adoption of EAI. Therefore, rather than generalising the outcome of these case studies, the author proposes to examine each case by describing respective approaches to the adoption of EAI and the evaluation of integration technologies. In doing so, allowing others to draw parallels in outcome.

5.1 Introduction

Chapters 2 and 3 of this dissertation have identified that there is a need to further investigate and analyse the adoption of enterprise application integration in organisations. The absence of theoretical models that focus on EAI adoption have led the author to propose a novel model that consists of eight factors that influence the adoption of EAI namely: (a) EAI benefits; (b) EAI barriers; (c) EAI costs; (d) IT infrastructure; (e) IT sophistication; (f) external pressures; (g) support and, (h) the existence of an evaluation framework that supports organisations to assess integration technologies. The author examines the validity of the proposed conceptual model using the case study strategy. In doing so, the cases of two multinational organisations are presented and analysed in the following sections. The author selected only two case organisations since they provided enough information for this research. Selecting of a third case company would give marginal benefits to this work.

5.2 Case Study One – The AUTOCORP

5.2.1 Background to the Organisation

Due to confidentiality reasons, the author uses the name *AUTOCORP* to refer to the organisation being reported. AUTOCORP is a multinational organisation that traditionally operates in the automotive sector. It has up to 200.000 employees in 132 countries and has an annual turnover of €31.6 billions. The organisation consists of 250 subsidiaries and affiliated companies in 50 countries. AUTOCORP has 185 production plants worldwide, with 43 of them located in its home-country with the rest remaining in Europe, Africa, Asia, Australia and North and South America. AUTOCORP also holds interest in 37 joint-venture companies around the globe. AUTOCORP is not only a name for automotive equipment such as driver information systems, ABS, brakes, and fuel-injection technology but also, for a whole range of further product areas. Examples are household appliances, automation technology, power tools, communications technology, thermo-technology, and packaging machinery. These worldwide activities of AUTOCORP are divided into four business units-sectors namely: (a) automotive equipment; (b) communication technology; (c) consumer goods and, (d) capital goods.

5.2.2 Background to Integration Problem

During the last decade, the tremendous changes in the business arena have pushed AUTOCORP to become more efficient and competitive. AUTOCORP believes that a flexible infrastructure is required to maintain and expand its business to a global level. Such an infrastructure will allow the organisation to easily adapt to its changing business environment and gain a competitive advantage.

Initially, AUTOCORP recognised the need for a flexible and manageable IT infrastructure when it attempted to address the year 2000 problem (Y2K) and migrate from its ERP SAP R/2 package to SAP R/3. The need for an integrated and flexible IT infrastructure has been necessitated with the existing infrastructure causing numerous problems to the organisation. These problems became an obstacle for AUTOCORP as they prevented it from implementing its business goals. For instance, AUTOCORP could not support its goal of closer collaboration and coordination of inter-organisational business processes due to the non-integrated nature of its applications. This held the organisation back from achieving competitive advantage and cost reduction. The main problems that were caused by the existing IT infrastructure are presented below and are classified in the same way as in Chapter 2 into technical, managerial and financial:

Technical Problems: The existing IT infrastructure is heterogeneous and consists of hundreds of incompatible systems. As a result, AUTOCORP *faces significant integration* problems when attempting to migrate its existing custom built applications to SAP R/3. Another problem was the incorporation of best-of-breed ERP modules to SAP R/3. Following a best-of-breed approach, AUTOCORP purchased the "best" ERP modules that were available in the market. This means that AUTOCORP combined modules from different vendors. Unifying these systems is a problem since most of these modules are incompatible. In addition, each module was customised in a unique way to communicate with other existing systems (e.g. legacy systems). As a result, it was difficult for AUTOCORP to reconfigure and piece together all best-of-breed modules that run on a mainframe based SAP R/2 to the non-mainframe based SAP R/3. In addition, there was a redundancy of data and functionality, as many applications store similar data or run systems that overlap in functionality. The reasoning is that each business unit or subsidiaries have their own IT infrastructure. Moreover, subsidiaries use programs with similar functionality to automate their business processes. In each subsidiary, applications were customised in a unique way (e.g. based on the financial law and regulations of the home-country). In most subsidiaries, many systems store data for the same entity (e.g. a specific customer), which results in data redundancy. The reason for this is that applications can not share common data or objects due to integration problems. Additionally, the non-integrated infrastructure causes many problems to the organisation since it could not achieve supply chain and eProcurement integration. Therefore, AUTOCORP could not take advantage of information technology and support closer collaboration with its suppliers and customers.

Financial Problems: AUTOCORP's suppliers and customers demand closer collaboration with the organisation. However, the insufficient IT infrastructure could not accomplish a tighter collaboration at both intra-organisational and inter-organisational level. This situation resulted in a lost of sales since AUTOCORP could not efficiently support its customers or coordinate its activities with its suppliers. Another important financial problem was the high operational cost of the existing IT infrastructure. AUTOCORP believes that it is not cost effective to support a large infrastructure, which includes numerous systems with overlapping functionality. The maintenance cost of such an infrastructure is also high, which presents an additional financial barrier. The organisation elaborated possible solutions to overcome this situation with one of the solutions that were proposed focusing on point-to-point interconnectivity. AUTOCORP estimated that in case of interconnecting existing applications, the costs of managing the new evolving interfaces would be tremendous. It estimated that the time to configure one interface will be about 15-20 man/days. This time will be much more since each interface should be altered when an interconnected system is changed. This indicates that point-to-point connectivity leads to extravagant solutions with expensive maintenance cost.

Managerial Problems: Also the limitations of their existing IT infrastructure cause problems in management. Since multiple applications store data for the same entity (e.g. a specific supplier) management could not retrieve the most updated data for this entity and therefore faced problems in decision-making. AUTOCORP requires flexible, cross-organisational core business processes, such as: (a) development; (b) controlling; (c) sales;

(d) quality management and, (e) finance and accounting, which must be based on a homogenous and flexible IT infrastructure. The latter will allow the organisation to be more flexible in adapting to the changes of the business environment. Existing IT infrastructures can not efficiently support core business processes and, is therefore, becoming an obstacle for achieving business goals. In addition, the strong need in the automotive industry for the integration of inter-organisational business processes requires the integration of new systems into existing infrastructures. In order to streamline business processes between the organisation and its trading partners, AUTOCORP uses eProcurement systems and online stores. Nonetheless, there is a need for better collaboration among trading partners by fully integrating the organisation. There is also a strong need to integrate SCM and CRM systems to improve coordination and relationships with suppliers and customers. However, the existing IT infrastructure can not support this requirement due to its non-integrated nature. All these problems are summarised in Table 5.1.

Integration Drivers	Problems
Technical	▪ Problems in migrating existing applications (legacy, custom built) to SAP R/3 ▪ Problems in incorporating best-of-breed ERP modules ▪ Problems in supporting supply chain management integration and eProcurement integration ▪ Problems in providing a homogeneous IT infrastructure ▪ Difficulties in maintaining the IT infrastructure ▪ Redundancy of data and applications ▪ Traditional interconnectivity approaches have a high complexity
Financial	▪ Existing infrastructure has a high operational cost ▪ Traditional interconnectivity approaches have a high cost ▪ Lost of sales
Managerial	▪ Existing infrastructure can not efficiently support management. ▪ The inability of existing infrastructure to provide data accuracy causes problems in decision making ▪ The non-integrated infrastructure was a problem for the collaboration and coordination of cross-enterprise and enterprise wide activities and processes. ▪ Problems in integrating intra-organisational business processes ▪ Problems to integrate business processes with customers and suppliers.

Table 5.1: AUTOCORP - Problems of the Non-integrated IT Infrastructure

The empirical data extrapolated from AUTOCORP has revealed many factors that stimulate the adoption of an integrated infrastructure. These factors include:

- external pressures such as increased competition and a requirement for closer collaboration with trading partners;
- the limitations of existing IT infrastructure;
- cost factors that are related with the maintenance of existing infrastructure; and,
- cost factors that are associated with the development of non-flexible and manageable point-to-point solutions.

5.2.3 Motivation for EAI Adoption

In the late nineties, the organisation started introducing the new version of SAP software (SAP R/3). The adoption of SAP R/3 requires the step-by-step migration of existing systems. As a result, AUTOCORP migrated legacy systems such as IBM CICS or Siemens BS2000 to SAP R/3. Also, SAP R/2 and other custom built solutions like EDI applications were migrated to SAP R/3. The migration of existing applications to SAP R/3 was not based on a standard architecture but rather, on custom point-to-point interconnections. Thus, as it is illustrated in Figure 5.1, the migration resulted in application spaghetti. As stated in earlier chapters of this dissertation (see Chapters 1,2 and 3) point-to-point interconnectivity causes many maintenance problems and resulted in non-flexible IT infrastructures.

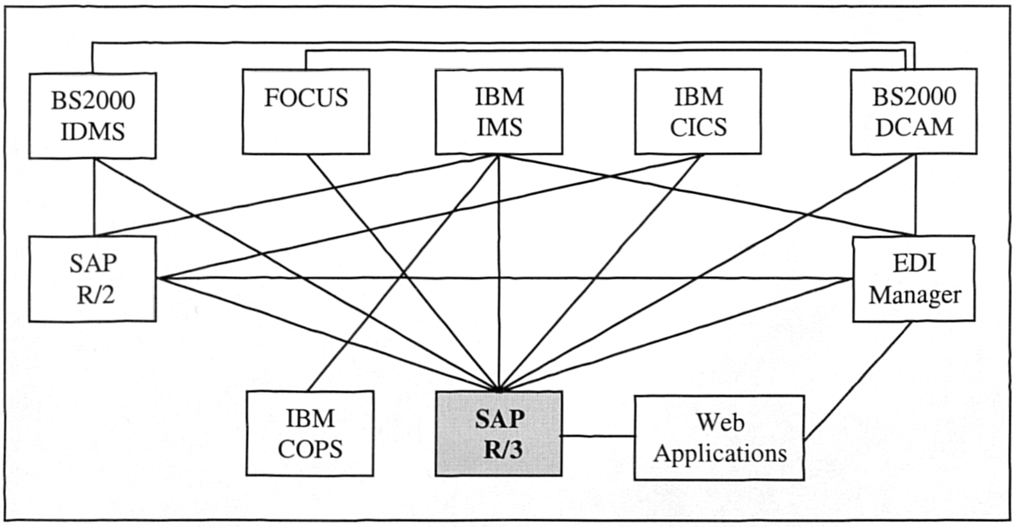

Figure 5.1: Applications Spaghetti at AUTOCORP

AUTOCORP estimated that the interconnection of SAP R/3 with the most common used existing application requires the programming and maintenance of up to 300 interfaces. Furthermore, the demand for closer collaboration and coordination with customers and suppliers requires the incorporation of e-commerce applications, supply chain management, customer relationship management and eProcurement applications. The organisation estimates that at a European level, a total of 700 new interfaces are required to interconnect these applications.

As a manufacturer, AUTOCORP uses SCM systems for the coordination and optimisation of its internal and external supply chains. The requirements evolving from the implementation of e-business systems, such as reduced cycle time and 'available-to-promise' checks, drive the need for a tighter integration between AUTOCORP's plants, central and regional warehouses and finally the suppliers. The availability of a new era of SCM systems enable AUTOCORP to develop an inter-organisational model for the supply chain to integrate more tightly its business processes with those of its suppliers. Therefore, SAP Advanced Planner and Optimiser (APO) system is currently projected for the retail business units. With the implementation of this system several custom built applications, SAP R/2 and SAP R/3 systems have to be integrated, which will result in about 200 new interfaces.

However, there was much scepticism in AUTOCORP regarding applications interconnectivity since it results in a non-manageable IT infrastructures. Finally, the organisation started seeking other possible solutions to incorporate its systems, with an External Consultant (E.C.) justifying this decision:

> *"Many reasons affected our decision to seek new ways of integrating the IT infrastructure. Firstly, the cost of interconnecting and maintaining these interfaces and applications is high. Secondly, point-to-point interconnections do not result in a flexible and manageable IT architecture and infrastructure but rather create more chaos and complexity. This complexity will be dramatically increased, as there is an increasing trend to incorporate many more applications with AUTOCORP infrastructure... The amount of interfaces will be rapidly increased."*

Another motivation for EAI adoption was the implementation of an e-business strategy, since it requires the integration of: (a) business processes and, (b) e-business, custom and packaged applications. AUTOCORP sees the integration of its intra-organisational business processes and the harmonisation of its internal IS architecture with SAP R/3 as a requirement for the integration of inter-organisational processes with its suppliers and customers through e-business applications. One of the first systems that AUTOCORP implemented was a solution for one of its subsidiaries. For confidentiality reasons the author uses the name AUTOCORP SUB_A to refer to this company[1]. AUTOCORP SUB_A is a manufacturer of car communication technologies, e.g. car audio, traffic telematics, and radiophones. AUTOCORP SUB_A is one of the European market leaders in car radios with a volume of five millions car radios per year. In July 1998, AUTOCORP SUB_A introduced an Internet-based electronic catalogue, which enables specialized traders and aftermarket customers to order products, and to obtain information about products, prices, delivery status and backlogs. AUTOCORP wants to unite the sell-side applications and catalogues under the same AUTOCORP Portal, to provide a corporate identity for its customers. As Figure 5.2 depicts, the portal shall be used to incorporate cross-business unit processes for both business-to-business and business-to-consumer. By providing an overall AUTOCORP portal, several ERP and custom-built applications, even from different business units, have to be integrated. There is therefore, a need for an integrated and standardized architecture to implement its e-business strategy.

[1] AUTOCORP SUB_A, AUTOCORP SUB_B, AUTOCORP SUB_C etc are names that are used by the author to refer to AUTOCORP subsidiaries.

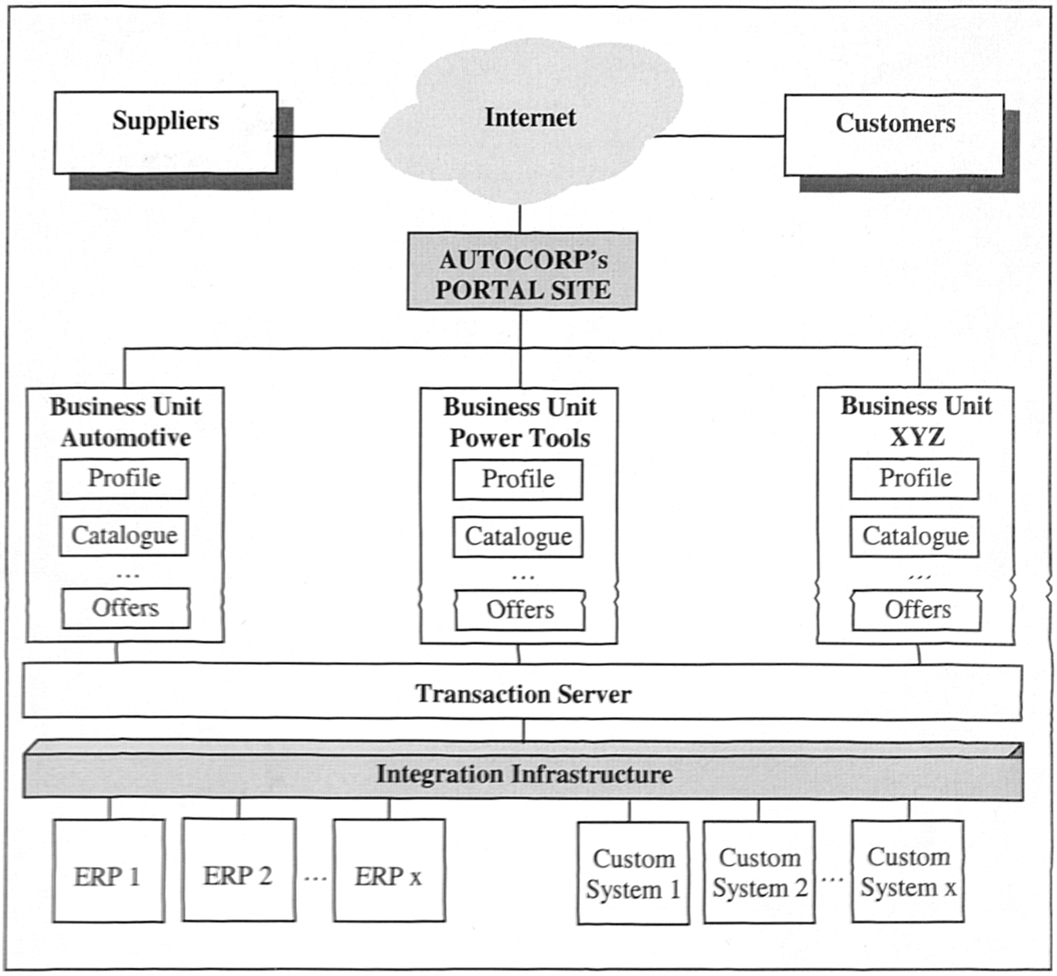

Figure 5.2: AUTOCORP's e-business Architecture

Corporate identity is not the only reason for an overall portal strategy. As some of the AUTOCORP customers order products from different business units, AUTOCORP wants to profit from the synergies of an integrated solution. The external consultant gave as an example of such integration the case of AUTOCORP SUB_B, which not only integrates AUTOCORP tighter with its customers, but also integrates different business units within AUTOCORP. AUTOCORP SUB_B buys car audio equipment from the business unit Power Tools of AUTOCORP SUB_A, and orders windscreen wipers and spark plugs from the business unit Automotive Aftermarket. AUTOCORP aims at providing a shop-in-shop portal solution that offers more transparent and streamlined processes for the customers. The project manager added that:

"This will allow us to become more customer oriented. Since customers have different needs in different regions, it is our goal to offer regionally individualized web sites for the specialised and wholesale trade as well as for the end-consumer... Because of the complexity that evolves from this implementation project, an integrated IT infrastructure will offer a more flexible approach to AUTOCORP."

In interpreting from the empirical data, it appears that internal pressures influence the adoption of enterprises application integration at AUTOCORP. It is revealed that strategic reasons (e.g. changes in business strategy) seem to be a factor for affecting the decision for introducing EAI at the organisation. Another factor that appears to influence the adoption of enterprise application integration is related to the limitations of other existing solutions (e.g. point-to-point interconnectivity).

5.2.4 EAI Adoption Process

AUTOCORP believes that it is a big challenge to bring together all information systems and fully automate the organisation. At the end of 1999, the IT department started examining available solutions to meet the challenge for developing a standardised, flexible, integrated and homogeneous IT architecture. In doing so, the IT department, prepared few feasibility studies based on the capabilities of existing integration technologies. After reviewing these studies, AUTOCORP took the decision that application integration presents a significant approach for developing a manageable and homogeneous IT infrastructure. It therefore, appears that IT sophistication influences the decision to adopt EAI technology.

However, AUTOCORP did not take the decision to fully integrate the organisation since such a solution has a high cost. This also indicates that cost factors also influence the decision for introducing EAI. In addition, the project manager reported that the plan for developing a global integrated IT infrastructure was considered of high risk for the following reasons:

- there is no single application integration technology or software package that supports the development of a global integrated IT architecture;

- there is a lack of knowledge in the organisation regarding incorporating applications based on EAI solutions; and,
- there is a need to fully reengineering the business processes so as to take advantage of EAI technology. However, business process redesign will affect many relationships at both intra-organisational and inter-organisational level, with AUTOCORP not being able to estimate this impact.

This indicates that barriers like the lack of knowledge of EAI and the lack of a single EAI product that solves all integration problems, influenced AUTOCORP's decisions regarding EAI adoption.

For all these reasons, the managing board took the decision to implement a pilot EAI project. The pilot project will integrate applications following a process centric scenario, and will incorporate business processes at a both intra-organisational and inter-organisational level. Based on such an EAI adoption approach, AUTOCORP will be able to justify the adoption of a global EAI project. The reason for this is that AUTOCORP will extract important observations and identify possible benefits and barriers. Therefore, the organisation will evaluate the efficiency and the risks of the pilot EAI project and thus, justify the adoption of a global EAI solution.

5.2.5 The IT infrastructure

AUTOCORP is a big multinational organisation that has 250 subsidiaries and affiliated companies in 50 countries. Each of these companies has its own IT infrastructure. As a result, the organisation consists of hundreds of incompatible and heterogeneous information systems. For instance, AUTOCORP consists of more than 2000 custom built systems that are based on a diversity of platforms, operating systems, data structures and computer languages. Most of these systems are legacy applications that run on mainframe environments such as IBM CICS and Siemens BS2000.

Since there was a lack of common IT infrastructure, and a lack of central coordination of IT, the majority of AUTOCORP subsidiaries adopted their own packaged-ERP system. The amount of ERP packages that exist in AUTOCORP can not be reported, as the interviewees did not specify the number of ERP systems in the organisation. Also, *important technical information was not given by AUTOCORP since it believes that the combination of its systems is a competitive advantage.* Similarly, the amount of e-business solutions was not specified. However, two consultants that were involved in this case study reported that AUTOCORP has more than 100 ERP installations and up to 90 e-business applications.

At a global level, AUTOCORP runs ERP software from a single vendor (SAP). The reason for using only SAP is that the organisation is satisfied from the overall functionality of the package. Furthermore, the usage of the same package at a global level results in reducing the heterogeneity in its IT infrastructure and thus, better supports management and reduces costs. For instance, employees' training is better organised by AUTOCORP and it costs less, as all subsidiaries use the same package. Moreover, the IT departments can share the obtained knowledge regarding SAP modules thus, reducing the costs of external consultants.

SAP software at AUTOCORP runs on both mainframe (SAP R/2) and non-mainframe (SAP R/3) environments. Although the organisation and its subsidiaries use the same ERP package there is no common installation, since each subsidiary or business unit has customised SAP modules in a different way. The SAP R/3 customisation was based on various parameters such as: (a) the business needs; (b) the strategy of each company and, (c) the local financial regulations.

The first installation of SAP R/3 took place in AUTOCORP's telecommunications division in Frankfurt, at the begging of 1995. Today, 20,000 AUTOCORP employees all over the world work with SAP R/3 every day in all divisions. These divisions include: (a) automotive equipment; (b) power tools; (c) household appliances; (d) thermo-technology; (e) automation technology and, (f) packaging machines. In some cases SAP R/3 was used to integrate IT departments by replacing existing systems. For instance, in AUTOCORP SUB_C in Belgium, the SAP R/3 integrated solution replaced a heterogeneous application landscape consisting of several single systems, some of which were linked by interfaces. However, such installation of SAP R/3 afforded only partial integration of the dataset management. In AUTOCORP SUB_C SAP R/3 went live with about 500 users on schedule at the beginning of May 1999, as part of an SAP project at AUTOCORP in Belgium. So far, AUTOCORP has been using SAP R/3 in the areas of Accounting, Logistics and Human Resources, deploying practically all of the classic SAP R/3 modules. Around the world, Windows NT is used as a server operating system in several "server farms" (about 150 servers in total). In May 2000 SAP's official Internet site (www.sap.com) reported that AUTOCORP has over 40 SAP R/3 systems.

AUTOCORP's 20,000 live SAP R/3 users represent one of several SAP milestones in the organisation, with the number of SAP R/3 users in AUTOCORP set to rise to over 50,000 in the medium term. IT projects instituted more recently are based on products in SAP's New Dimension initiative, namely the SAP Advanced Planner and Optimiser (SAP APO), SAP Business Information Warehouse (SAP BW) and SAP Business-to-Business Procurement (SAP BBP) in conjunction with SAP R/3.

At an European level, an interdisciplinary information-processing department supports AUTOCORP, and coordinates the activities of the IT department at all business units. For confidentiality reasons, the author uses the name IT SUPPORT to refer to this information-processing department. With IT SUPPORT, AUTOCORP has a dedicated "SAP Pool", an internal team of consultants who deal specifically with SAP R/3 implementation and application issue. The pool's members combine specialist knowledge of SAP R/3 with general business know-how, therefore cover not only all relevant SAP topic areas, but also ensure that standardized master data, basic structures and templates are used. As well as carrying out project work and safeguarding AUTOCORP's process knowledge, the pool employees ensure a continuous know-how transfer between projects, locations, and business areas, and between AUTOCORP and SAP.

As illustrated in Figure 5.3, IT SUPPORT shares its knowledge with AUTOCORP's business units, and provides them with support in order to coordinate the functionality of SAP.

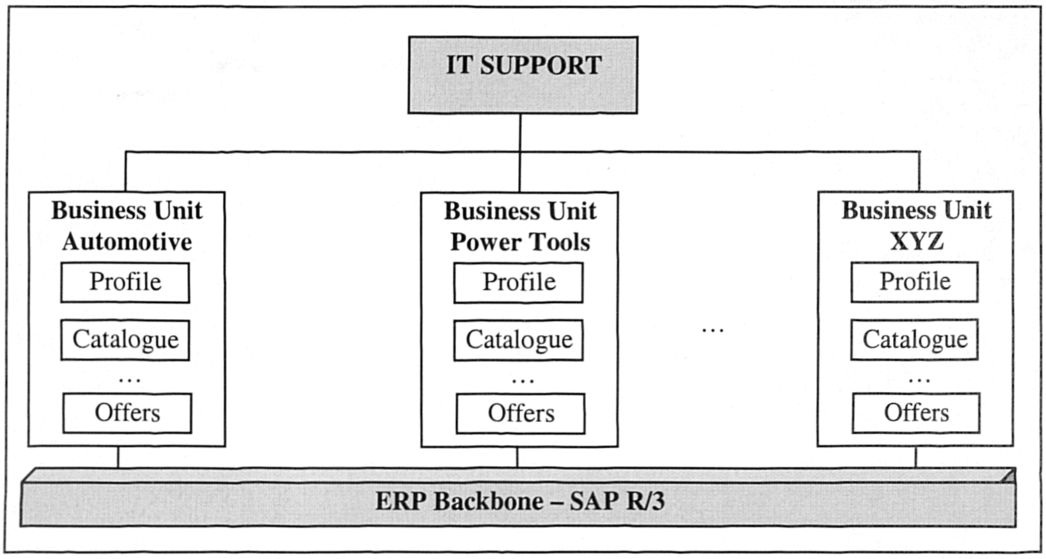

Figure 5.3: Association of AUTOCORP's Business Units to IT SUPPORT

The customers and suppliers demand for a closer collaboration with the organisation has led AUTOCORP to develop an infrastructure that supports a tighter model of cooperation. This infrastructure interconnects a number of e-business applications such as SCM, CRM, eProcurement and product lifecycle management with existing applications (e.g. SAP R/2, SAP R/3, custom built systems etc). As a result, AUTOCORP provides an interconnected e-business infrastructure to its customers and suppliers. As illustrated in Figure 5.4, such an infrastructure was developed using bespoke point-to-point interconnections. Therefore, resulting in applications spaghetti and thus, presenting various limitations as it has already been discussed in section 3.1.

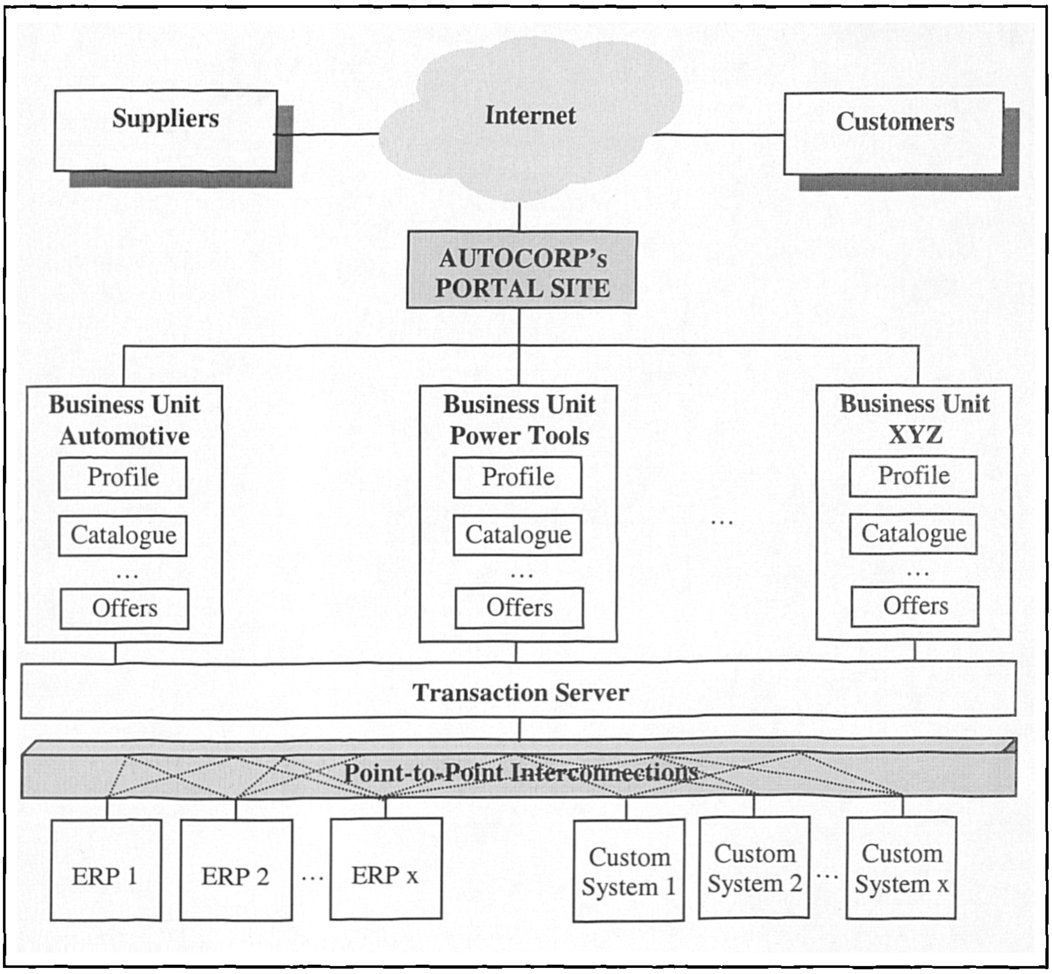

Figure 5.4: AUTOCORP's e-business Architecture based on Point-to-Point Interconnections

5.2.6 Evaluating Integration Technologies

Before proceeding to the implementation of the pilot EAI project, the IT department of AUTOCORP took the decision to evaluate integration technologies and products. As reported by project manager, AUTOCORP believes that there is no single EAI product that covers all integration requirements of its IT architecture. To decide which of the available EAI products is suitable for the AUTOCORP business bus (the proposed integration infrastructure for AUTOCORP), the company evaluated five different EAI products from vendors like: (a) BEA Systems; (b) CrossWorlds; (c) IBM; (d) Level 8 Systems and, (e) Mercator Software. Among others, the evaluation process focuses on the following criteria:

- **Integrated Vs Toolkit application:** With the implementation of SAP R/3, AUTOCORP needs to integrate several mainframe and custom-built applications, for which no standard *adapters* can be used. Additionally, AUTOCORP already uses IBMs MQ Series for the physical transport of data between applications. Therefore, the EAI product must be like a toolkit application, which: (a) allows the developers to individually build *adapters* for in-house developed systems and, (b) can be used with existing tools that are already used as a standard for application integration.

- **Tightly Vs Loosely coupling:** The *connectivity* services of an EAI product enable data integration by using synchronous and asynchronous mechanisms. By applying one of these mechanisms, it can be differentiated if the EAI solution supports a more tightly or a more loosely coupling of applications. EAI products that support a tightly coupled application assist companies with synchronous integration, whereas EAI solutions that support loose coupling assist companies with asynchronous integration. Similarly to the majority of other organisations that have adopted EAI solutions, both synchronous and asynchronous mechanisms are applied in the case of AUTOCORP. A prominent example for asynchronous data integration is the exchange of master and transactional data between distributed ERP systems. In contrast, synchronous mechanisms are very often used for e-business applications, which support available-to-promise checks in the ERP systems. As AUTOCORP needs a flexible architecture that supports both asynchronous and synchronous integration scenarios, the company needs an EAI product that supports both.

- **Individual Vs Standard application integration:** A major component of EAI products are the interface services that provide functionality for the translation of different application's APIs and object models. Most of the EAI vendors, such as CrossWorlds and Level8 Systems, have concentrated on APIs and object models of standard business applications like SAP R/3, Oracle and Baan. Only few vendors like BEA or Mercator Software that originally built traditional middleware solutions deliver EAI products that support the customer with functionality for the integration of legacy systems such as IBM CICS or Siemens BS2000. Especially for historical grown, multinational companies like AUTOCORP that have already used legacy systems, the support of such systems is crucial as the replacement of them is often not profitable. The reason for this is that many of these legacy systems: (a) are reliable; (b) handle critical applications [in the case of AUTOCORP] (e.g. manufacturing systems); (c) the replacement of these systems will be risky and cost high amounts of money; (d) their functionality is difficult to be replaced by other systems, since they are specialised and, (e) there is no justification, time and money for their replacement.

- **Intra Vs Inter-organisational integration:** The integration of intra and inter-organisational business processes and applications was another evaluation criteria for AUTOCORP. The organisation requires an EAI product that supports the integration of both intra and inter-organisational applications and processes. The evaluation of EAI products indicates that EAI vendors with strong middleware background such as BEA Systems and Level 8 Systems support intra-organisational EAI. The relatively new vendor-players in the EAI market like CrossWorlds have specialised in providing e-business integration and therefore, support more inter-organisational EAI. CrossWorlds offers standard e-commerce process configurations, which can be customized easily with a graphical workflow-modelling tool. Mercator Software has a background in the integration of EDI and back end systems. This is the explanation for Mercator policy to integrate a wide range of EDI standards and scenarios such as UN/EDIFACT and Odette in its Mercator product suite.

Apart from the evaluation criteria presented in Table 5.2, AUTOCORP used many other criteria. However, the organisation did not share these criteria for confidentiality reasons. When the project manager was asked to comment he reported that:

> *"The confusing nature in the integration marketplace requires employees with EAI skills and integration technologies. Currently, there is a shortage of skilled employees... our company has spent money and time to acquire this knowledge. We believe that enterprises that have this knowledge [IT sophistication on EAI]*

are in a position to adopt EAI. The sooner you adopt this knowledge the better."

It appears that AUTOCORP had difficulties in understanding integration technologies due to the confusion in the integration marketplace. It seems that an evaluation framework supported the organisation to adopt EAI, since it improved IT sophistication and allowed AUTOCORP to understand the capabilities of integration technologies and EAI packages. Table 5.2 summarises the evaluation results.

		Evaluation Criteria			
		Integrated Vs Toolkit Application	**Tightly Vs Loosely Coupling**	**Individual Vs Standard EAI**	**Intra Vs Inter-Organisational EAI**
EAI Vendor	**EAI Product**				
BEA Systems	elink	Toolkit Application	Tightly and Loosely Coupling	Individual EAI	Intra-Organisational EAI
CrossWorlds	United Applications Architecture	Integrated Application	Tightly Coupling	Standard EAI	Intra and Inter-Organisational EAI
IBM	MQ Series Integrator	Toolkit Application	Loosely Coupling	Individual EAI	Intra-Organisational EAI
Level 8 Systems	Enterprise Integration Template	Toolkit Application	Tightly and Loosely Coupling	Standard EAI	Intra-Organisational EAI
Mercator Software	Mercator	Toolkit Application	Loosely Coupling	Individual EAI and SAP R/3	Intra and Inter-Organisational EAI

Table 5.2: AUTOCORP - Evaluation of EAI Packages

Based on the evaluation results, AUTOCORP took the decision to adopt a variety of EAI products to integrate within the organisation. This proves that none of the EAI packages evaluated meets all evaluation criteria set by the organisation. Considering that the five EAI packages assessed represent the elite of EAI solutions (Ring and Ward-Dutton, 1999) fit appears that there is no single EAI package that addresses all integration problems. This finding is in accordance of other literature findings (Linthicum, 2000b; 2001; Ring and Ward-Dutton, 1999; Ruh *et al.*, 2000). For that reason, the IBM MQ Series integrator was used for the messages brokering. BEA elink was selected to support tightly and loosely coupling and Mercator software to facilitate inter and intra-organisational application integration. In addition, all three products can be used as toolkit applications and individual EAI. For confidentiality reasons, AUTOCORP members refused to provide more information regarding the applicability (range and level of use) of each product in the organisation.

Evaluation criteria set by AUTOCORP are similar to the criteria proposed by Ring and Ward-Dutton (1999) for the assessment of EAI packages (see Chapter 3). The author maps AUTOCORP's criteria to those criteria proposed by Ring and Ward-Dutton (1999) for the evaluation of EAI packages and summarises them in Table 5.3.

Evaluation Criteria used by AUTOCORP	**Evaluation Criteria proposed by Ring and Ward-Dutton (1999)**	**Description**
Integrated Vs Toolkit	Application Vs Toolkit	Toolkit criterion describes whether an EAI package can be used as an out-of-box product. In this case integrators have no understanding about the technical details of the package but they just know what the package does. Although toolkit packages ideally refer to *total* EAI solutions there is no such solution available today (Ring and Ward-Dutton, 1999). Application or Integrated EAI packages are tool-based solutions, which are accessible by integrators who can use the tools to upgrade and enhance the system.
Tightly Vs Loosely	Tight Vs Loose	Tight and loose criteria refer to the connectivity mechanism that EAI packages support. In most cases organisations require both types of mechanisms with loose integration related to asynchronous communication and tight to synchronous communication (see section 2.3.2)
Individual Vs Standard EAI	Custom Vs Packaged EAI	This set of criteria tests whether an EAI package focuses on the integration of custom (individual) systems or packaged (standard) applications.
Intra Vs Inter-organisational EAI	Internal Vs External EAI	Likewise, internal and external EAI focuses on the integration of intra or inter-organisational systems and business processes. In most cases, enterprises require both types of integration.

Table 5.3: Evaluation Criteria for the Assessment of EAI Packages

It appears that both AUTOCORP and Ring and Ward-Dutton (1999) assess the capabilities of EAI packages to support the:

- integration of system types (custom, packaged, inter-organisational [e-business]);
- type of integration (loose, tight) and,
- availability of EAI packages that can be configured individually or used as a toolkit.

The last criterion indicates that EAI packages consist of a set of tools that are based on integration technologies reported in Chapter 3. This criterion tests whether EAI packages allow integrators to customise these technologies-tools (e.g. adapters) based on their own needs.

5.2.6.1 Scope and Analysis of the Proposed Evaluation Framework for EAI Technologies

The following sub-section contributes towards the assessment of the novel evaluation framework that was proposed in section 3.5. In achieving this, those evaluation criteria considered to support the assessment of integration technologies are identified, when seen from a multiple-stakeholder view. These views were seen from those stakeholders that were involved in the adoption, evaluation and implementation of EAI, as it was not possible to interview all AUTOCORP stakeholders. The stakeholders that were interviewed included: (a) an External Consultant (E.C.); (b) an integrator (Int.) and, (c) the Project Manager (P.M.) of the project. Both, integrator and the project manager, work for the Germanic AUTOCORP where the external consultant is based in Switzerland and collaborates with the German and the Swiss AUTOCORP. The author interviewed the aforementioned stakeholders, as it was not possible to interview all AUTOCORP stakeholders. The interview agenda reported in Appendix C was used during these structured interviews.

Interviewees were asked to identify the importance of the evaluation criteria and then, to assess the integration technologies using the four categories of evaluation criteria (see Table 3.3). The level of importance presented in the tables of this chapter follows a scale similar to the one used by Miles and Huberman (1994) with the values: (a) low importance; (b) medium importance and, (c) high importance presented by ○, ◒ and ●, respectively. In those cases where the interviewees did not report a level of importance, the author uses the mark '-' to show this reaction.

This sub-section summarises and analyses interviewees' perceptions related to integration technologies and issues. Also this section presents their comments regarding the proposed framework for evaluating integration technologies. Table 5.4 summarises interviewees' perceptions when asked to identify the importance of the integration requirements.

In interpreting the empirical data, it appears that interviewees share similar perceptions regarding the integration requirements. Nearly all integration requirements presented in Table 5.4 are considered of great importance when incorporating applications. However, there are a few minor differences in interviewees answers dealing with: (a) scalability; (b) product maturity; (c) technology maturity and, (d) custom-to-custom application integration.

When the integrator was asked to comment, his answers regarding product and technology maturity he said:

> *"The maturity of a product or a technology is not so important for us... I will give an example to understand my point. Ten years ago, the Internet emerged as a new technology that supported enterprises and allowed them to take advantage [of Internet technology]. Those companies like amazon.com that proactively adopted the Internet have gained a competitive advantage. Many people have criticised the Internet for its maturity and security but it has finally become a technology that changed the whole business and technological environment. The same happened with other technologies like Java, XML and so on."*

Interviewees do not share the same perceptions regarding the integration of custom-to-custom applications. External consultant reported that this is of low significance with project manager and integrator saying that it is of medium and high importance, respectively. When the author asked the project manager to explain his answer he said that:

> *"We had the impression that custom-to-custom applications integration was not so important but, there are cases in which we have to perform this type of integration."*

Integration Requirements	E.C.	Int.	P.M.
Maintainability	●	●	●
Flexibility	●	●	●
Scalability	◒	●	◒
Portability	●	●	●
Reusability	●	●	●
Product Maturity	◒	○	◒
Technology Maturity	◒	○	◒
Low Complexity	●	●	●
Non-invasive	●	●	●
High Performance	–	●	●
Real Time Integration	●	●	●
Mainframe Compatibility	●	●	●
Non-Mainframe Compatibility	●	●	●
Support of Data Integration	●	●	●
Support of Objects Integration	●	●	●
Support of Process Integration	●	●	●
Support of Transportation Layer	●	●	●
Support of Transformation Layer	●	●	●
Support of Process Automation Layer	●	●	●
Support of Custom-to-Custom Application Integration	○	●	◒
Support of Custom-to-Package Application Integration	●	●	●
Support of Custom-to-e-business Application Integration	●	●	●
Support of Packaged-to- Packaged Application Integration	●	●	●
Support of Packaged -to-e-business Application Integration	●	●	●
Support of e-business-to- e-business Application Integration	●	●	●
Support of Custom-to-Packaged-to- e-business Application Integration	●	●	●
Other: Security	●	●	●
Other: Manageability	●	●	●
Other: Vendor Global Presence	–	–	●
Other: Vendor Support	–	●	●

Table 5.4: Identification of Integration Requirements at AUTOCORP

On the other hand, the external consultant reported that custom-to-custom applications incorporation is not important for AUTOCORP, since this type of integration is required in only a few cases.

Interviewees at AUTOCORP reported that security, manageability, vendor support and vendor's global presence should be considered as additional integration requirements. These requirements were also reported by literature (Ring and Ward-Dutton, 1999; Ruh *et al.*, 2000; Zahavi, 1999). Thus, it seems that these requirements should be taken into consideration when evaluating integration technologies, since there are both practical and literature evidences. The requirements that deal with vendor support and vendor's global presence are related to support factors. It appears that vendor's support influences the adoption of an integrated solution. Since, AUTOCORP has insufficient knowledge regarding EAI packages, it seeks for vendors that can support them (e.g. technical support).

Then interviewees were asked to evaluate the integration technologies using the four categories of criteria identified in section 3.4. Table 5.5 presents interviewees' perceptions regarding integration technologies when they assessed them using applications' elements as evaluation criteria.

	Application Elements								
	Data			**Objects**			**Process**		
Integration Technologies	**E.C.**	**Int.**	**P.M.**	**E.C**	**Int.**	**P.M.**	**E.C**	**Int.**	**P.M.**
ODBC	●	●	●	○	○	○	○	○	○
JDBC	●	●	●	●	◒	◒	○	○	○
RPC	●	●	–	○	○	–	◒	○	–
MOM	●	●	●	○	◒	○	○	○	○
Message Broker	●	●	●	◒	◒	○	◒	●	●
XML	●	●	●	●	●	●	●	◒	–
TPM	–	◒	◒	○	○	○	●	◒	–
Application Serves	●	●	●	●	●	◒	●	◒	○
CORBA	◒	●	◒	●	●	●	◒	○	–
DCOM / COM	◒	◒	◒	●	●	◒	◒	○	–
EJB	○	○	○	◒	◒	●	◒	○	○
Screen wrapper	◒	◒	○	○	◒	○	○	○	–
APIs	●	◒	◒	◒	●	●	○	○	○
Adapters	●	●	●	●	●	●	○	○	–

Table 5.5: AUTOCORP-Evaluating EAI Technologies Using Application Elements as Criteria

Clearly, the evaluation results show that there is no single technology that supports the integration of all applications' elements. This is in accordance with literature findings like

Sharma *et al.*(2001) and, indicates that a combination of technologies is required to facilitate the integration of data, objects and processes. When the project manager was asked to comment his answers he said:

> *"Each technology partially supports the integration of data, objects and processes. Some of these technologies such as message brokers and adapters are more powerful solutions than others... It is difficult to say which is the best using this table [Table 5.5]. First of all we have to understand the applicability of each technology and that's why we have to map them against integration layers... Integration layers allow us to see which technologies support a layer. In each layer we have to seek for technologies that support all applications elements"*

Thereafter, interviewees were asked to assess integration technologies using the second category of evaluation criteria (integration layers) with Table 5.6 presenting evaluation results.

The interviewees reported that practically message brokers are not used to support transportation layer although they can support it. This is attributed to that developers preferring to use message brokers for the translation and process automation layer and adopt other technologies for transportation layer. In addition to the aforementioned integration layers, interviewees consider *Connectivity* as an integration layer. When an external consultant was asked to explain more this perception, he said:

> *"We consider connectivity as an important integration layer. This layer [connectivity] is responsible for creating the connections-interfaces among the applications and the central integration infrastructure. Through these connections application elements are passed from one system to the transportation layer. Then transfers these elements to the central integration infrastructure where transformation and process automation are taken place."*

	Integration Layers								
	Transportation			Transformation			Process Automation		
Integration Technologies	E.C.	Int.	P.M.	E.C	Int.	P.M.	E.C	Int.	P.M.
ODBC	◒	○	○	○	●	◒	○	○	○
JDBC	◒	○	○	○	●	◒	○	○	○
RPC	◒	◒	–	○	○	○	○	○	○
MOM	●	●	●	○	○	–	◒	○	–
Message Broker	●	●	●	◒	●	●	●	●	●
XML	●	●	●	●	●	●	◒	◒	–
TPM	◒	◒	○	○	○	–	◒	–	○
Application Serves	◒	◒	◒	●	◒	–	●	○	–
CORBA	◒	◒	○	◒	◒	◒	◒	◒	○
DCOM / COM	◒	◒	◒	◒	○	○	◒	–	–
EJB	○	◒	◒	○	○	○	◒	○	–
Screen wrapper	○	○	○	○	◒	◒	○	○	○
APIs	○	○	○	○	◒	◒	○	○	–
Adapters	○	○	○	●	●	●	–	◒	○

Table 5.6: AUTOCORP-Evaluating EAI Technologies Using Integration Layers as Criteria

Table 5.7 shows the assessment of integration technologies when interviewees used the third category of evaluation criteria (system types). Based on their answers, it appears that message brokers support the integration of all system types. This is in line with both the literature (Linthicum, 2000a; 2001; Ring and Ward-Dutton, 1999; Sharma *et al.*, 2001) and practice with EAI vendors using message brokers as the main integration engine of their EAI solutions. Adapters and XML appear to support all or nearly all system types. However, the integrator mentioned that:

> *"This table [Table 5.7] is broad as it does not maps the integration technologies against the integration layers. Many technologies support the integration of various system types but some of them support one layer and some other another. What we have to do is categorising these technologies using integration layers to understand what is going on."*

Interviewees were then asked to assess these technologies using integration requirements as evaluation criteria with Tables 5.8 and 5.9 summarising their answers. Clearly, the results show that there is no single technology that meets all evaluation criteria with message brokers fulfilling nearly all criteria. However, interviewees mentioned that organisations should not focus on one or another category of evaluation criteria when assessing integration technologies but, take all of them into consideration. More specifically external consultant said that:

> *"All sets of criteria you gave [he means the proposed framework] are too important for the evaluation of integration technologies. I believe that organisations have to consider all these criteria and assess technologies in a similar way."*

In addition, interviewees found the proposed framework very helpful and they reported that it improves IT sophistication and influences the decision for EAI adoption. The reasoning is that the proposed framework supports decision-making and allows the IT departments, to better understand the capabilities of integration technologies, as well as their integration requirements. Moreover, they express their intention to adopt the proposed framework.

Integration Technologies	System Types																				
	Custom-to-Custom			Custom-to-Packaged			Custom-to-e-business			Packaged-to-packaged			Packaged-to-e-business			E-business-to-e-business			Custom-to-Packaged-to-e-business		
	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.
ODBC	●	◐	○	○	◐	◐	○	◐	○	○	●	◐	○	●	◐	○	●	◐	◐	●	◐
JDBC	●	○	○	○	◐	◐	○	◐	○	○	◐	○	○	●	◐	○	●	●	◐	◐	◐
RPC	-	◐	-	◐	◐	-	○	◐	-	◐	○	-	○	○	-	○	○	-	○	○	-
MOM	◐	◐	◐	◐	◐	◐	○	◐	○	◐	◐	◐	○	◐	○	○	○	○	◐	◐	○
Message Broker	●	●	●	◐	●	●	◐	●	●	◐	●	●	◐	●	◐	○	◐	◐	◐	●	●
XML	○	○	○	●	○	○	●	●	●	●	◐	◐	●	●	●	●	●	●	◐	●	◐
TPM	○	◐	-	○	○	-	○	○	○	◐	○	○	○	○	○	○	○	○	○	○	○
Application Serves	-	○	○	●	◐	◐	●	●	◐	◐	◐	◐	●	●	●	●	●	●	◐	◐	◐
CORBA	●	◐	-	◐	◐	○	○	◐	◐	○	●	◐	○	●	●	○	●	●	◐	◐	◐
DCOM / COM	○	○	○	◐	◐	○	○	◐	-	○	●	◐	○	●	◐	○	●	◐	◐	●	◐
EJB	○	○	○	○	○	○	○	◐	◐	○	◐	●	○	◐	●	○	●	●	○	◐	◐
Screen wrapper	○	●	●	○	●	◐	○	●	◐	○	○	○	○	○	○	○	○	○	○	◐	○
APIs	○	○	○	●	◐	◐	○	○	◐	●	●	●	○	●	●	○	◐	◐	◐	◐	◐
Adapters	◐	◐	◐	◐	◐	◐	◐	◐	◐	●	●	●	◐	●	●	○	◐	◐	◐	●	●

Table 5.7: AUTOCORP-Evaluating EAI Technologies Using System Types as Criteria

Integration Technologies	Integration Requirements																	
	Maintainability			Flexibility			Scalability			Portability			Reusability			Maturity		
	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.
ODBC	◒	◒	–	◒	○	–	○	◒	○	○	●	◒	◒	●	●	●	●	●
JDBC	●	◒	◒	●	◒	◒	○	○	○	●	◒	◒	●	●	◒	●	◒	◒
RPC	○	○	–	○	○	–	○	–	–	○	○	–	○	○	–	●	●	–
MOM	◒	◒	○	◒	◒	◒	◒	◒	○	◒	○	–	◒	○	○	●	●	●
Message Broker	●	●	●	◒	●	●	●	●	●	◒	●	●	●	●	●	●	◒	○
XML	●	●	●	●	●	●	○	◒	◒	●	●	●	●	◒	–	◒	◒	◒
TPM	○	○	–	○	○	○	●	◒	○	○	◒	●	○	–	–	●	●	●
Application Serves	◒	●	◒	◒	◒	●	●	◒	●	◒	●	●	◒	◒	–	◒	◒	◒
CORBA	●	◒	◒	◒	◒	◒	◒	◒	◒	●	◒	–	●	●	◒	●	●	●
DCOM / COM	◒	◒	●	◒	◒	◒	◒	◒	–	◒	○	○	●	●	●	◒	●	●
EJB	●	◒	●	◒	◒	●	○	◒	–	◒	◒	–	●	●	●	○	○	○
Screen wrapper	○	○	○	○	○	○	○	◒	○	○	○	–	○	○	○	●	●	●
APIs	○	◒	◒	○	◒	◒	○	○	○	○	○	○	○	◒	◒	●	●	●
Adapters	●	●	●	●	●	●	◒	◒	◒	●	◒	◒	●	◒	◒	◒	○	○

Table 5.8: AUTOCORP-Evaluating EAI Technologies Using Integration Requirements as Criteria

Integration Technologies	Integration Requirements																	
	Complexity			Non-invasive			Performance			Real-Time			Mainframe Compatible			Non-Mainframe Compatible		
	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.	E.C.	Int.	P.M.
ODBC	○	◒	–	◒	◒	–	○	◒	○	◒	◒	○	○	○	○	●	●	●
JDBC	○	○	–	◒	–	–	○	○	○	◒	◒	○	○	○	○	●	◒	●
RPC	●	●	–	–	○	–	○	○	–	●	●	–	◒	●	–	●	●	–
MOM	◒	●	●	–	○	○	◒	◒	○	◒	○	○	◒	●	●	●	●	●
Message Broker	◒	○	○	●	●	●	●	●	●	◒	◒	–	◒	●	●	●	●	●
XML	◒	◒	◒	◒	◒	◒	○	◒	◒	○	◒	○	◒	○	○	●	●	●
TPM	●	●	–	–	○	○	◒	●	●	◒	–	○	●	◒	○	◒	●	●
Application Serves	●	◒	–	–	○	–	●	◒	–	◒	◒	–	◒	○	○	●	●	●
CORBA	●	◒	◒	–	○	–	◒	◒	◒	●	◒	–	●	●	●	●	●	●
DCOM / COM	◒	○	○	–	○	–	◒	◒	◒	●	●	–	◒	○	○	●	●	●
EJB	●	–	–	–	○	–	○	○	○	◒	●	●	◒	○	○	●	●	●
Screen wrapper	●	●	●	–	●	–	○	○	○	◒	○	–	●	●	●	◒	◒	◒
APIs	●	●	●	–	◒	◒	○	○	–	●	●	●	◒	◒	◒	●	–	–
Adapters	○	○	○	–	●	◒	◒	◒	–	◒	●	–	●	●	●	●	●	●

Table 5.9: AUTOCORP-Evaluating EAI Technologies Using Integration Requirements as Criteria

5.2.7 Pilot EAI Project

The aim of the pilot project was to prove that application integration could be used for the development of a standardised, flexible and maintainable infrastructure that integrates both intra and inter-organisational business processes and applications. For that reason, the pilot project attempted to test whether EAI supports a robust IT infrastructure that achieves: (a) closer collaboration with customers and suppliers and, (b) better coordination of business processes. Another target of the pilot project was to demonstrate possible benefits and highlight barriers of application integration. In doing so, it would help the IT department and managing board justifying the adoption of a global EAI solution. In interpreting from empirical data, it appears that AUTOCORP considers EAI benefits and barriers as factors that influence its adoption.

The pilot project was started in May 2001 and finished 6 months later (October 2001). It was designed to incorporate custom and packaged applications integration. The reasons for this decision were that:

- AUTOCORP consists of a vast amount of custom systems (more than 2000);
- Packaged systems such as SAP R/3 'govern' the overall functionality of the organisation, as the majority of important processes run on packaged systems;
- Most e-business modules are designed to collaborate with other existing systems and therefore, are easier to be pieced together; and,
- AUTOCORP has recently implemented an interconnected infrastructure (as mentioned in section 5.2.5) that supports e-business applications. There is no justification to run another pilot project that provides similar functionality.

One of the main objectives of the pilot project was to increase coordination in demand planning. Therefore, the pilot project was designed to integrate seven business processes among business units and another five processes at inter-organisational level (AUTOCORP, customers and suppliers). These processes are summarised in Table 5.10.

Intra-organisational processes	Inter-organisational processes
Series sale	Customer Relationship Management
Sales samples	Supplier Relationship Management /eProcurement
Development/Product Data Management	Supply Chain Management
Sales Planning & Distribution	Collaborative Product Commerce
Controlling	Business Management
Pricing	
Guarantee & Quality Management	

Table 5.10: Business Processes that were Integrated during AUTOCORP's EAI Pilot Project

The project was developed at a European level and during its implementation a number of employees were involved including: (a) staff from the IT departments of AUTOCORP and its business units; (b) internal consultants; (c) external consultants; (d) IT SUPPORT and, (e) staff from AUTOCORP's suppliers and customers that participate in the pilot phase. Apart from the technical staff participated in the project a number of managers from all involved companies and business units had an important role in the project. The reason for this is that the pilot project was based on process centric integration, which requires the incorporation of both applications, and common business processes of all participants (AUTOCORP, AUTOCORP's customers and suppliers). Therefore, the organisation did much business process reengineering with its customers and suppliers. AUTOCORP estimated that the 70% of its overall time that was spent in the project deal with system design and business process reengineering.

At a technical level, application integration was adopted to piece together AUTOCORP's customers and suppliers with its business units. In doing so, the organisation developed an integration infrastructure that was called Business Bus. As illustrated in Figure 5.5, the business bus integrates the SAP R/3 system with custom-built systems that deal with material management. At an inter-organisational level, the business bus incorporates systems that are based at AUTOCORP's suppliers and customers and are used to automate common business processes. SAP R/3 and its module that supports Advanced Planner Optimiser (APO) function in an integrated way, since SAP R/3 is an integrated suite. This means that all SAP modules are internally integrated with the core system. Also, APO is unified with material management and other systems (e.g. customers) through the business bus.

Figure 5.5 depicts the configuration of one business unit that uses the pilot EAI infrastructure. Internally the advanced planner optimiser functions in an integrated way, which means that: (a) demand planning; (b) production planning and detailed scheduling; (c) deployment; (d) global ATP and, (e) supply network planning, are all pieced together and share common data. The global ATP sub-module communicates with SAP R/3 and retrieves data from other modules such as sales, orders and inventory control. These modules are continuously updated with data that are provided by customers and suppliers (e.g. an order). Data that are retrieved by global ATP are then forward to APO sub-modules (e.g. production planning, deployment) and support demand planning in analysing and optimising data. Moreover, APO and/or SAP R/3 modules exchange and/or retrieve data from other applications (e.g. material management, customer applications) that are significant for the functionality of APO or SAP R/3.

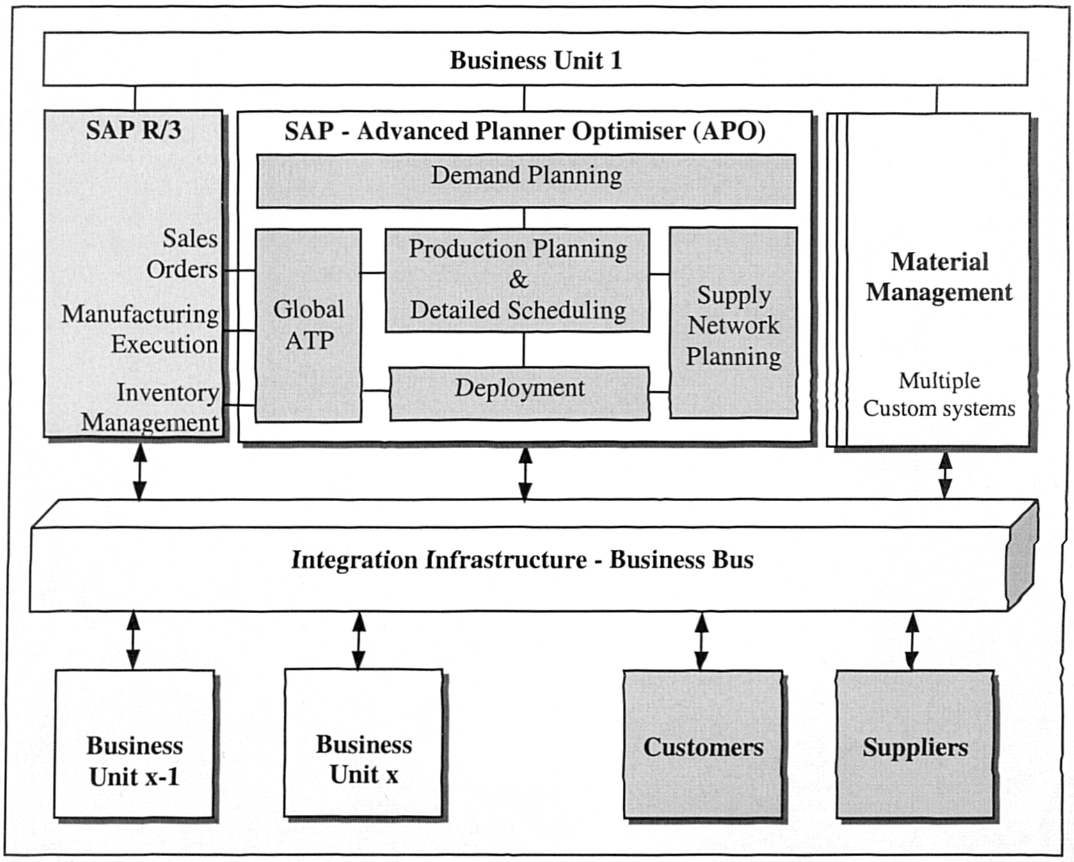

Figure 5.5: AUTOCORP's EAI Pilot Project – Integration Configuration for one Business Unit

As mentioned earlier in this section, the integration scenario was based on a process centric approach. This approach governed the whole integration efforts since integrators should incorporate all parts of the same process that run on a diversity of systems. As a result, integrators started piecing together the first part of a process that runs on one system (e.g. the process *orders* that runs on SAP R/3) and then incorporate the next logical part of the same process that runs on another system. This task was repeated sequentially, until all parts of the same process were unified through the integration of business bus.

Figure 5.6 presents the overall pilot application integration architecture in which multiple business units are integrated with multiple customers and suppliers.

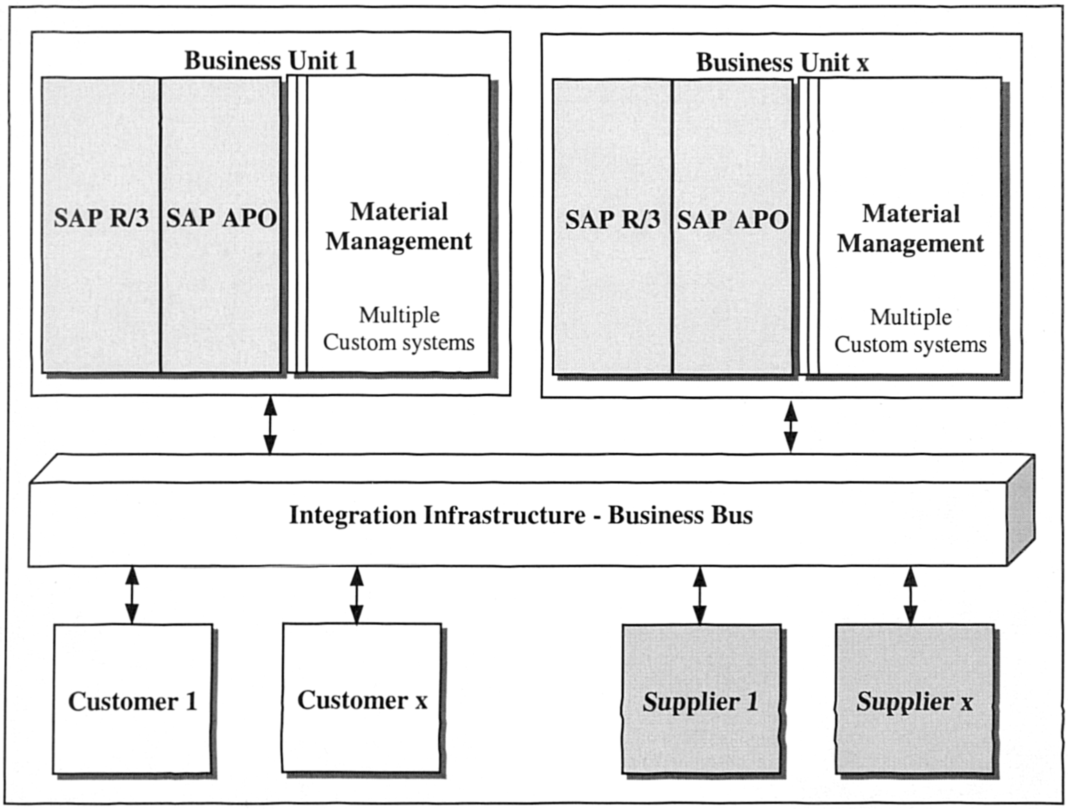

Figure 5.6: AUTOCORP's EAI Pilot Project – The Integrated Infrastructure

5.2.8 Global EAI project

After the completion of the pilot project, the IT department presented its results to the managing board. The latter analysed the benefits, barriers, the technical solution and the costs from the adoption of the pilot EAI project, and took the decision to integrate the organisation at a global level. When the project manager was asked to report the main reasons for this decision he summarised them into the following:

- Application integration supports a best-of-breed approach, which allows AUTOCORP to efficiently piece together the diversity of ERP modules that exist in the organisation;
- An EAI solution provides more flexibility at intra and inter-organizational process integration and change of business processes across different heterogeneous information systems;
- The integration business bus architecture provides a backbone for the flexible integration of suppliers and customers with customer relationship management, supply chain management and e-business applications;
- An EAI solution increases collaboration and coordination between AUTOCORP, its trading partners and affiliated companies. Also, application integration results in customer and supplier satisfaction and achieves competitive advantages;
- Although, application integration has a high adoption cost, it reduces the overall operational and maintenance costs, which is translated into a cost effective solution. In addition, EAI results in cost savings in implementing new and maintaining existing applications; and,
- Barriers to application integration adoption such as employees' resistance to change are important. It is AUTOCORP policy to overcome EAI drawbacks and barriers and take advantage of this new technology. Global competition pushes enterprises to adopt more effective and flexible ways of doing business and thus, AUTOCORP will move forward by implementing an EAI infrastructure.

This shows that the decision for EAI adoption at AUTOCORP was influenced by the following factors: (a) EAI benefits; (b) EAI barriers; (c) EAI costs; (d) the technological solution that EAI supports; (e) increased competition and, (f) strategic factors (e.g. gaining competitive advantage). However, as mentioned in previous sections the adoption decision at AUTOCORP was not limited to these factors (e.g. another factor is the use of the evaluation framework).

The global EAI project was started on December 2000 and will have a 6-year duration. AUTOCORP estimates that by December 2006 the organisation will be able to function worldwide in an integrated way. In doing so, the organisation will incorporate all types of information systems (custom, package and e-business applications) by integrating all permutations of system types (e.g. custom-to-packaged, custom-to-e-business).

The integration is based on a process centric approach, and it is divided in three main phases namely:

- **Design.** Although, this phase is expected to be the longest in the overall project, its duration can not be precisely estimated. The project manager claims that the design of an integrated IT infrastructure is complicated, and requires much more time than the design of a traditional application. However, the amount of time required for design can not be identified since there are not enough studies deal with the EAI design phase. Designers and system analysts at AUTOCORP believe that design is the most significant phase of the project since, it reviews, redesigns and improves all business processes. At this phase, the work effort needed is enormous since all business processes, information systems and relationships should be re-examined and be understood by business analysts and system designers. Design is a phase of high importance, since if mistakes occur during this phase, these mistakes will affect the whole project as well as the enterprise and/or cross-enterprise business processes and relationships. As a result it will increase the overall implementation time and cost.

- **Implementation of mega datacentre.** A new phase that deals with the development of a global mega datacentre was justified to take place after design and before the implementation of the EAI solution. To standardise and improve data quality, business analysts and designers believe that they should develop mega datacentres at global and regional levels. These mega datacentres will store and handle data in a harmonised way and better support management in taking decisions. The reason for this is that all the latest data will be stored in a single datacentre and thus, managers will be able to retrieve and easily analyse data. AUTOCORP estimates that custom-built systems will be dramatically reduced and therefore, mega datacentres should be built around SAP R/3. The reason for this is that: (a) SAP R/3 handles great amounts of data and, (b) SAP R/3 will be the core system for AUTOCORP at a global level. The organisation is planning to implement a global mega datacentre to consolidate finance and accounting. In doing so, AUTOCORP will be based on SAP R/3 modules for finance and accounting. In addition, the organisation will develop regional datacentres to harmonise data (e.g. orders, sales) at a regional level and support the collaboration with customers and suppliers.

- **Integration business bus implementation.** The last phase of the global application integration project involves the implementation of the system. As illustrated in Figure 5.7, business units are integrated at both global and regional level. At regional level, each business unit has an EAI infrastructure that integrates all applications at intra-organisational level. This infrastructure supports the incorporation of custom, packaged and e-business applications integration. Using its regional EAI infrastructure, a business unit coordinates its business processes with other business units, customers and suppliers that belong to the same region (e.g. Europe Union). Likewise, a business unit uses regional systems (like regional SAP R/3 for production and sales) to fulfil its tasks. A business unit uses its regional and global business bus to: (a) store or retrieve data from the global mega datacentre and/or, (b) collaborate with a business unit or a trading partner in another region. Figure 5.7 depicts a regional EAI architecture where Figure 5.8 illustrates the global application integration infrastructure at AUTOCORP.

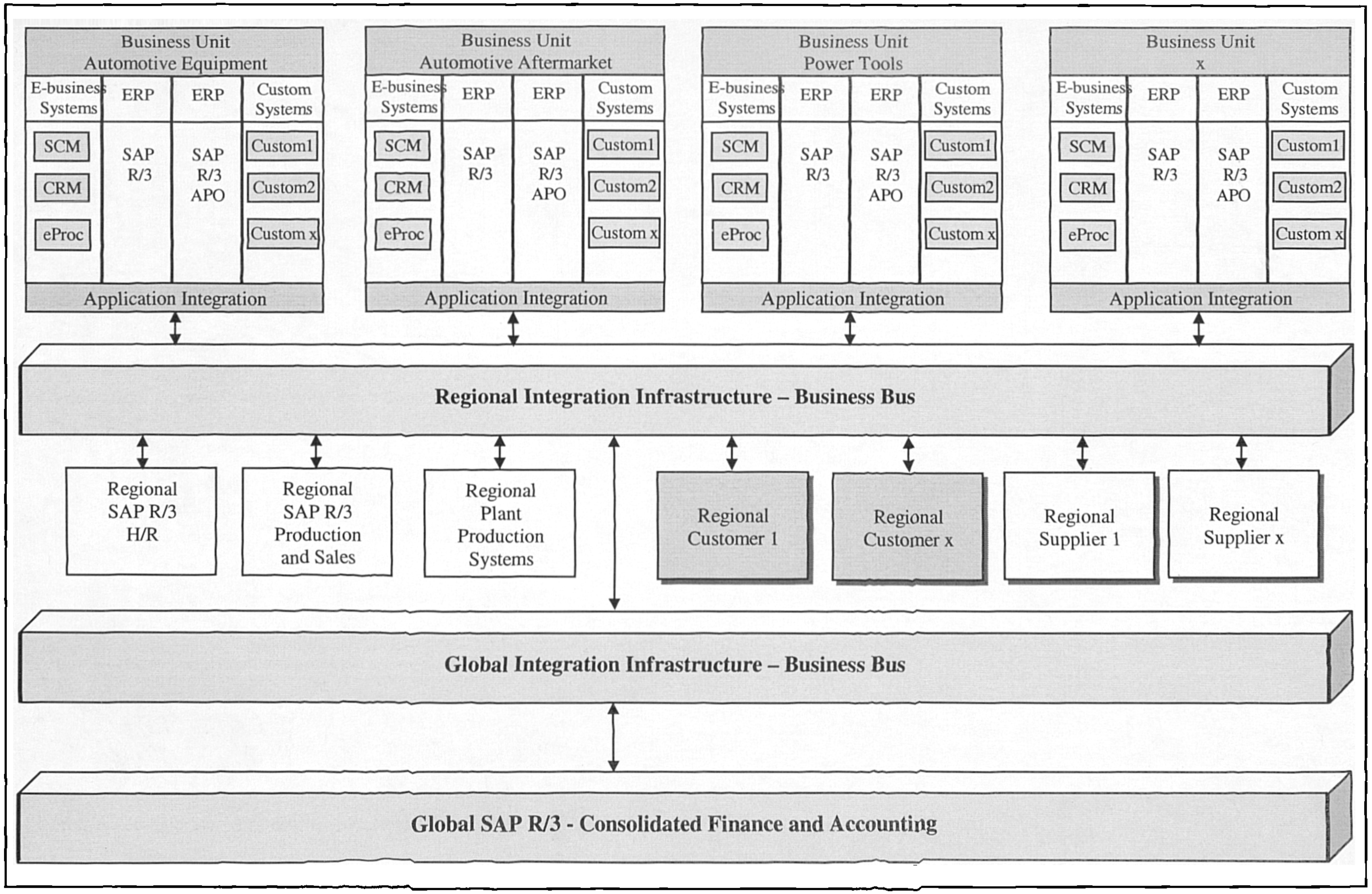

Figure 5.7: AUTOCORP's Global EAI Project – Regional Integrated Architecture

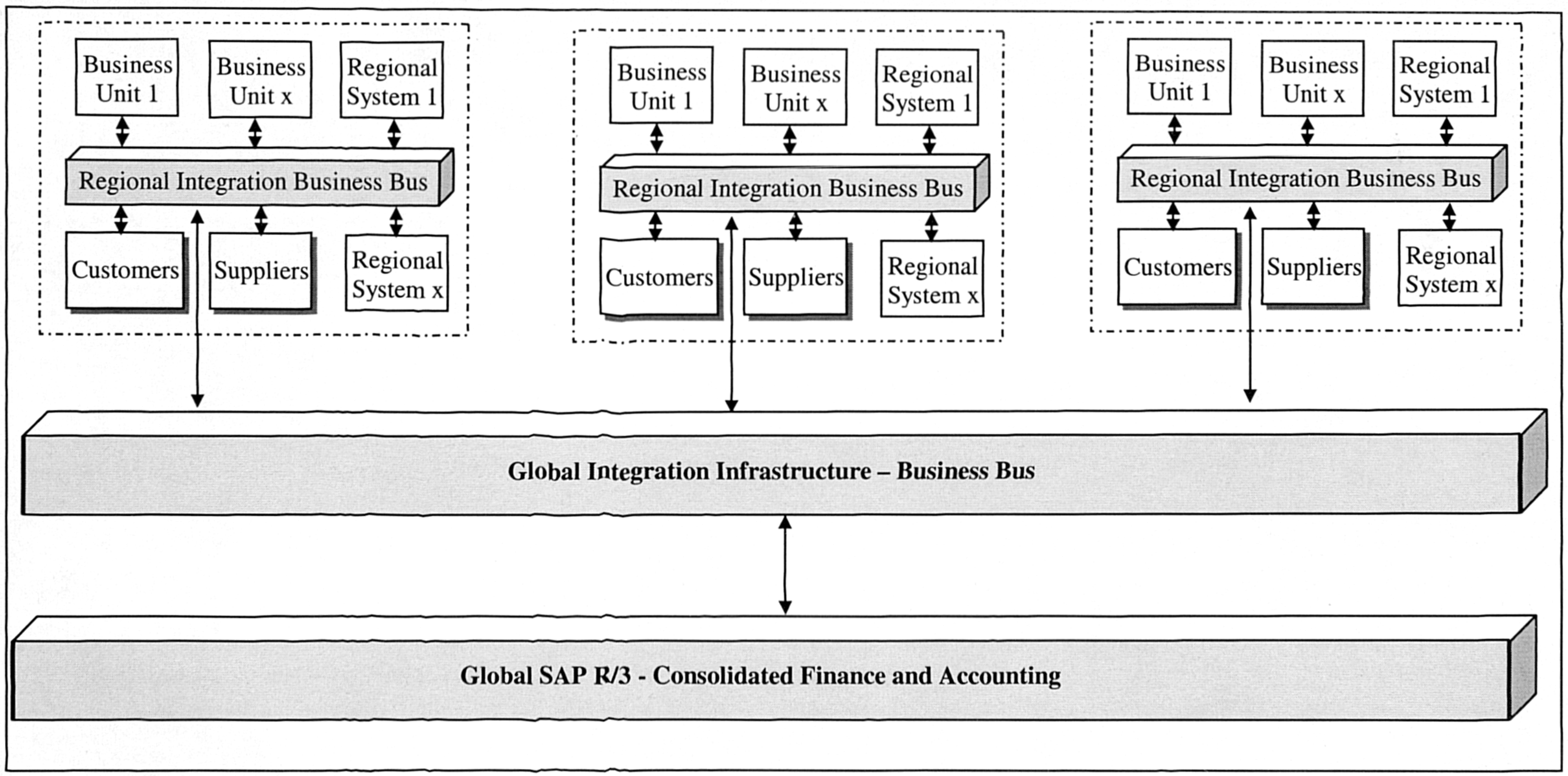

Figure 5.8: AUTOCORP's Global EAI Architecture

5.2.9 Benefits

The author asked the interviewees to determine the benefits from the implementation of an Integrated IT infrastructure. All interviewees agree with project manager who reported that:

> *"Although we identify the benefits from the adoption of pilot EAI project, we can not generalise these benefits and say that we will have the same benefits after the implementation of the global EAI project... We do not expect a big variation between the benefits of the pilot and the global EAI project. For example, the pilot project may result in a 10% of cost reduction where the global project in an 8.5%."*

Table 5.11 presents interviewees' answers regarding the benefits from the adoption of an integrated IT infrastructure. Table 5.11 categorises EAI benefits according to the model proposed by Shang and Seddon's (2000) (see section 2.5.1). It appears that interviewees share common perceptions regarding EAI benefits. The main findings include:

- **Operational benefits:** EAI reduces the cost of managing, running and maintaining the IT infrastructure. The integration has an impact on reducing the overall operational cost at AUTOCORP due to business process reengineering and organisational restructuring. This has also resulted in increased productivity since processes have been optimised and fully automated and integrated. Another operational benefit deals with planning improvement in supply chain management since internal and external supply chains are integrated through EAI.

- **Managerial benefits:** The process reengineering that has taken place during the pilot EAI project has resulted in more organised business processes, as well as allowing AUTOCORP to better understand and control its processes. In doing so, business processes have been improved which resulted in increased performance. In addition, the integration has resulted in data quality, which also improves performance and management as it supports decision-making process through the integrated infrastructure.

- **Strategic benefits:** The project manager reported that the pilot EAI project has achieved a 232% Return On Investment (ROI). This is in line with other published case studies (Anonymous, 1999a; 1999b; 1999c) that report ROI between 200% and 300%. In addition to this, interviewees mentioned that the pilot EAI project has achieved customer satisfaction and resulted in increased collaboration among partners. The integrated IT infrastructure has resulted in customers' satisfaction since customers demanded tighter relationships with AUTOCORP. The development of an integrated IT infrastructure has also resulted in closer collaboration with customers and suppliers.

- **Technical benefits:** At a technical level many benefits were identified including: (a) flexible, manageable, maintainable IT infrastructure; (b) reduced data and systems redundancy; and. (c) faster and cheaper implementation than bespoke solutions etc. The EAI solution supported AUTOCORP in reducing the redundancy of data and systems. Less data and applications result in less maintenance effort. The use of EAI technologies for the development of integrated IT infrastructure has achieved a more flexible and manageable solution. Connections among systems are non-invasive which reduces maintenance tasks (as explained in Chapter 3).

- **Organisational benefits:** The integrated IT infrastructure that has resulted from the pilot EAI project has allowed the organisation to do business more effectively. This is attributed to business processes are being organised and fully automated and integrated. Thus, manual tasks have been dramatically reduced and unnecessary or redundant tasks have been eliminated.

Category	Application Integration Benefits	E.C	Int.	P.M.
Operational	Improves planning in supply chain management	●	●	◒
	Increases productivity	◒	◒	●
	Reduces cost	●	●	◒
	Quicker response to change	●	●	●
	Reduces lost sales	●	◒	◒
Managerial	Provides more understanding and control of processes	◒	◒	●
	Results in more organised business processes	◒	●	●
	Increases performance	◒	◒	●
	Improves data quality	◒	●	●
	Improves management and supports decision making	●	●	●
Strategic	Increases collaboration among partners	●	●	●
	Achieves customer satisfaction	◒	-	●
	Achieves return on investment	●	-	●
Technical	Offers interfaces-standardisation	●	●	●
	Results in reusable systems, components and data	●	●	◒
	Reduces redundancy of applications, data and tasks	◒	◒	◒
	Faster and cheaper implementation than bespoke solutions	●	●	●
	Increases flexibility	●	●	●
	Provides flexible, maintainable and manageable solutions	●	●	●
	Provides portability	●	●	●
	Reduces development risks	●	◒	◒
	Achieves non-invasive solutions	●	●	●
	Achieves process integration	●	●	●
	Increases data analysis	◒	●	●
	Supports efficient data sharing	●	◒	●
	Results in reliable data	◒	●	●
	Process and systems scalability	◒	◒	●
Organisational	Allow organisations to do business more effectively	◒	●	●

Table 5.11: Classification of AUTOCORP's EAI Benefits

5.2.10 Barriers

The adoption of a pilot EAI project has impacted on AUTOCORP. As illustrated in Table 5.12 interviewees reported that the introduction of EAI at AUTOCORP has a low political impact. The reason for this is that during the last years IT SUPPORT has coordinated and controlled all business processes and data that are automated through the SAP packaged.

Business units did not cause problems to the organisation when they were asked to share and integrate their data and processes. The reason for this was that business units were familiar with the idea of sharing their processes and data. It can be said that this is a cultural issue that deals with the management of business units.

According to interviewees the most important barriers to EAI adoption at AUTOCORP deal with the:

- high cost of business process reengineering and organisational restructuring;
- high complexity in understanding the business processes;
- complexity in improving and fully automating business processes;
- confusing nature of integration marketplace and the lack of a single product achieves integration; and,
- lack of employees with EAI skills.

Table 5.12 summarises and classifies the barriers according to the model proposed by Shang and Seddon (2000).

Category	Application Integration Barriers	E.C	Int.	P.M.
Operational	Extra cost for redesign and change business structure, processes	●	●	●
Managerial	High complexity in understanding the processes and systems in order to redesign and integrate them	●	●	●
	Complexity of business processes	●	●	●
	Earlier approaches on EAI had proved problematic	○	◒	●
Strategic	Politics issues	◒	–	●
	Political impact (e.g. who controls the processes)	○	○	○
Technical	No single EAI product solves all integration problems	●	●	●
	Lack of employees with EAI skills	○	●	◒
	EAI has a high cost	●	◒	●
Organisational	Resistance to change	●	●	●
	No time for training employees on integration technologies	○	●	◒
	Cultural issues	○	○	○

Table 5.12: Classification of AUTOCORP's EAI Barriers

5.2.11 Costs

When the author asked interviewees to identify the costs of EAI adoption at AUTOCORP they were reluctant to answer due to confidentiality reasons. Nonetheless, interviewees identified the importance of costs without giving specific amounts for their case. A discussion with an external consultant has led to the conclusion that the global EAI solution at AUTOCORP will cost tens of millions Euros.

According to Hochstrasser (1992), Irani (1998) and Irani *et al.* (1997, 1998) costs can be divided into direct and indirect cost factors. Direct costs are financial tangible and can be attributed to the implementation and operation of IT costs. Such costs may include initial hardware and software costs, maintenance costs (e.g. licenses, hardware and software maintenance), system development costs etc. Indirect costs are financially tangible/intangible and non-financial in nature and can be divided into indirect human costs and indirect organisational costs. Indirect human costs can include employee training, employee motivation, management effort and dedication where indirect organisation costs may include business process reengineering, losses in productivity, strains on organisation resources, organisational restructuring etc. Based on this analysis, Table 5.13 illustrates a taxonomy of the costs and Table 5.14 classifies EAI costs at AUTOCORP based on the aforementioned taxonomy (see Table 5.13)

Direct Costs	**Indirect Human Costs**	**Indirect Organisational Costs**
Hardware costs	Employees training	Business process re-engineering
Software costs	Changing employees culture	Organisational restructuring
Development costs	Management efforts	Covert resistance
Maintenance costs (hardware and software)		Strategy redesign
Consultancy costs		

Table 5.13: Taxonomy of Direct and Indirect Costs

As it is summarised in Table 5.14 interviewees reported that consultancy costs and business process reengineering were the highest cost during EAI pilot project. Software and development costs as well as employees training, strategy redesign and changing employees culture are reported as costs of medium importance. Hardware and maintenance costs are characterised as low costs.

Category	Application Integration Costs	E.C	Int.	P.M.
Direct Costs	Hardware costs	○	○	○
	Software costs	◒	◒	◒
	Development costs	◒	◒	◒
	Maintenance costs	–	○	○
	Consultancy costs	●	●	●
Indirect Human Costs	Employees training	◒	–	◒
	Changing employees culture	◒	–	◒
	Management efforts	–	–	◒
Indirect Organisational Costs	Business Process re-engineering	●	●	●
	Organisational restructuring	–	–	●
	Covert resistance	–	–	◒
	Strategy redesign	◒	–	◒

Table 5.14: Classification of AUTOCORP's EAI Costs

5.3 Case Study Two – The OILCORP

5.3.1 Background to the Organisation

The name of the organisation that is being reported can not be published due to confidentiality reasons. As a result, the author has adopted the name *OILCORP* to refer to this organisation and reflect its business sector. OILCORP is a multinational petroleum company with more than 100,000 employees operating in more than 135 countries worldwide. The company is organised into five core business divisions including:

- oil;
- gas and power;
- chemicals;
- renewables; and,
- exploration and production.

A Chief Executive Officer (CEO) heads each core business and he has broad overall responsibilities. The CEOs report to a committee of managing directors made up of executive directors serving on the boards of the parent company.

In implementing its business strategy, OILCORP has merged and acquired subsidiaries during the recent years. Subsidiaries operate independently but, they comply with the same set of business principles. The service companies provide a range of specialist advice and resources, and principles to ensure that all companies perform to the same high level in the economic, environmental and social domains.

5.3.2 Background to Integration Problem

Each subsidiary has its own IT infrastructure, which is operated and coordinated independently. The non-integrated nature of OILCORP's IT infrastructure has caused many problems to the organisation. Following the same level of analysis as in Chapter 2 these problems are classified into: (a) technical; (b) managerial; (c) financial and, (d) strategic.

- **Technical problems:** The organisation consists of hundreds of custom applications and tens of ERP systems. Obtaining data from custom systems is difficult, as the majority of these systems have incompatible and heterogeneous data structures and formats. In addition, there are many compatibility problems when retrieving data from ERP systems. Although, the majority of ERP systems were purchased from two ERP vendors, OILCORP has difficulties in retrieving and processing data, as these systems are running on various platforms that have different software versions. As reported by interviewees, SAP software presents compatibility problems when attempting to integrate different SAP versions. For instance, the organisation has problems in retrieving data from a SAP module running on a mainframe X.400 and processing them in another SAP system running on different platform or has different software version. In addition, there are restrictions from both custom and ERP systems as they are not able to manipulate all types of data due to the reasons reported in Chapters 1 and 3.

- **Managerial problems:** The diversity of information systems causes delays in giving information, as applications are not integrated, and much work has to be carried out manually. For instance, data from one system has to be printed out and then re-entered in a different format to a target system. The reason for this is that the target system has its own data structure and/or it is based on different operating systems. Nonetheless, the delays in delivering information cause problems in decision-making and management. The non-integrated infrastructure leads managers to inefficient decisions important information is often missing and/or data can not be retrieved from applications. For example, there is often a delay in sending information dealing with products availability. Therefore, in many cases the management can not take accurate decisions regarding the replacement of products. Thus, this inability leads to lost of sales and low customers satisfaction.

- **Financial problems:** OILCORP has realised that the non-integrated nature of systems cost the organisation money and time. This is attributed to the organisation having to spend high amounts of money to support and maintain all these systems. Additionally, the inability of OILCORP to efficiently serve customers has an extra cost as it leads to loss of sales and thus, customers often turn to competitors.

- **Strategic problems:** The amount of subsidiaries and/or the diversity of systems involved in serving clients has resulted in 'no single face to customers'. For example, various types of customers' data are required to support systems with the same functionality. As a result, customers should provide each subsidiary with different data types and data to fulfil similar processes. Only a few systems require the same data to perform the same functions around the organisation. In addition, the delays in giving information and 'no single face to customers' have also resulted in low customer satisfaction (as explained in previous paragraphs). When asked one of the Managers of Projects and Solutions (MPS) to comment on this issue, he said:

  "This situation [non-integrated IT infrastructure and no single face to customers] makes our group less efficient and competitive. We estimate that in the long or medium term this situation may lead our customers to

competitors. This will affect our group, which will loose part of its market and this is against our strategic targets."

5.3.3 Motivation for EAI adoption

During the last decade, the tremendous changes in the global business arena have led OILCORP to adopt e-business practices and applications (e.g. eSupply chain management) to gain competitive advantages. However, the adoption of e-business applications was not enough to allow OILCORP to achieve its targets. The reason for this is that the rapidly changing business environment, requires organisations to support flexible and manageable IT infrastructures to gain competitive advantages. In this context, OILCORP recognised that integration is a significant parameter that influences the success of e-business applications and supports it in achieving a competitive advantage. Traditional approaches to inter-organisational integration such as electronic data interchange have proved insufficient and complex for OILCORP. As reported by all interviewees EDI has a high cost and requires altering target and source applications. This results in non-flexible and manageable solutions. In addition, the nature of EDI standards in use (e.g. UN/EDIFACT) is complicated, which adds additional complexity.

Other approaches to integration such as ERP systems have failed to support OILCORP's intra and inter-organisational integration, since they co-exist alongside other applications. The fragmentation in ERP implementations across OILCORP is a constraint for successful e-business transformation. The reason for this is that, there are less than 100 ERP implementations at OILCORP. Many of them run over mainframes or they and do not support real-time capabilities. In addition, there are many compatibility problems among ERP systems (e.g. they do not support the same data formats or they were customised in a different way). Thus, the co-ordination of all these systems under an e-business umbrella that requires real-time data is an obstacle. Therefore, organisation believes that the way forward is to develop an integrated IT infrastructure by redesigning their IT infrastructure and phase out all redundant systems and data. In doing so, the complexity will be dramatically reduced and the organisation would based on a more flexible and integrated IT infrastructure.

The non-integrated IT infrastructure has caused OILCORP many problems since: (a) it has a high cost of maintenance; (b) it is not manageable; (c) it is not flexible; (d) it results in insufficient decision-making and, (e) leads to low customer satisfaction. In addition to this, there was a need to change the traditional asset-driven supply chain and become customer-driven value chain. As Figure 5.9 depicts this introduces a lot of changes in the organisation (e.g. *customers needs* guide (drive) the whole chain whether in a traditional supply chain suppliers and in-house core competencies are the one that guide (drive) the chain).

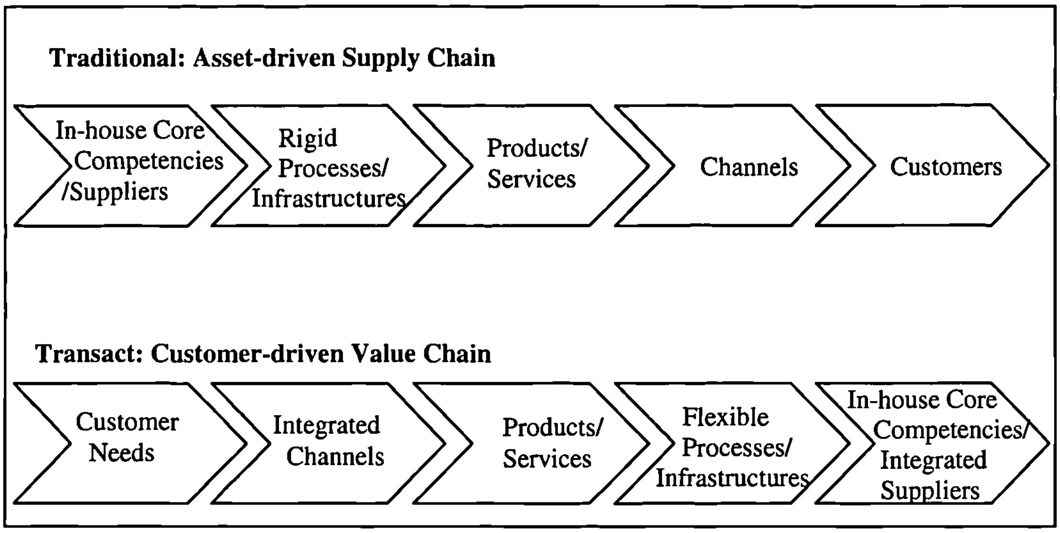

Figure 5.9: Asset-Driven Vs Customer-Driven Value Chain (Source OILCORP)

As illustrated in Figure 5.9 a customer driven value chain requires among others, integrated channels, flexible processes and infrastructures and integrated suppliers. However, the existing organisational structure and the IT infrastructure do not support such a transformation. Therefore, there is a need for rapid transformation from closed internal processes to open externalised processes. However, this target can be achieved through the development of an *integrated, adaptive and consisted IT* infrastructure across OILCORP.

5.3.4 EAI Adoption Process

The problems reported in the sections 5.3.2 and 5.3.3 and the cost of maintaining and running existing systems, have all led the organisation to seek a more efficient solutions for their IT infrastructure. In addressing these problems, the IT director asked internal consultants to search for available solutions. This action by IT director is in accordance with the *managerial perspective* reported in the normative literature for the adoption of new technology (Montealegre, 1998). Such approach supports, that the actions of managers are the primary causes of change and technology adoption (Montealegre, 1998). Managers attempt to meet internal policies, which often focus on increasing productivity, reducing costs and achieving competitive advantages. In doing so, managers bring computer resources into the organisation and distribute them throughout.

Internal consultants after reviewing a number of existing and new technologies (e.g. EAI) and studying similar cases, were persuaded that application integration could provide a significant solution by efficiently integrate the organisation. During informal conversations an internal consultant said that:

> *"We believe that EAI is the ultimate solution to our problems. With EAI we can develop an efficient, flexible and maintainable IT infrastructure by avoiding point-to-point interconnections."*

Internal consultants proposed the development of a global integrated IT infrastructure to integrate and automate most business processes in OILCORP. The proposed approach for integrating the organisation was based on a strategic mode[2] (Themistocleous and Irani, 2002) of EAI adoption. Such mode of integration would provide a solution to existing problems and, help the organisation to achieve its targets (e.g. developing a unified and manageable IT infrastructure). Internal consultants discussed this approach with an IT director who took the responsibility to introduce it to the managing board. Although, the idea for integrating the organisation was interesting, the managing board rejected it for the reasons reported below.

Previous integration attempts were not successful and there was a preconception regarding integrated solutions. The organisation had invested up to €812.6 millions [£500 millions] in less than 10 years to adopt ERP systems, which claimed to integrate and automate business processes.

However, once every year, the IT department proposed a new scenario/approach for automating businesses processes and thus, sough funding regularly. Although, the managing board supported the proposals for ERP adoption, none of them provided a total solution to their problems. As reported by all interviewees, this is attributed to that, ERP systems present many difficulties in collaborating with other existing or new applications.

Board members believed that existing systems were too complicated to be integrated. The reasoning is that such systems were: (a) based on different operating systems; (b) had a heterogeneous and incompatible nature; (c) did not support common standards and, (d) were not developed to collaborate with other applications. The proposed EAI solution was estimated to cost €250 millions [£153.8 millions], which was too high and therefore, the managing board could not risk this considerable amount of money.

After failing to introduce an integrated infrastructure, both IT department and internal consultants revised their initial proposal for the development of a global integrated IT infrastructure, using EAI. Their belief was that a strategic mode of EAI adoption would provide a significant solution to OILCORP problems, and leads them to propose three application integration pilot projects. Through these pilot projects, the IT department could evaluate application integration, demonstrate EAI benefits and decide whether EAI should be adopted by the organisation. Moreover, the IT department sought to solve existing technical problems (e.g. ERP systems redundancy) through the proposed EAI pilot projects. The cost of the implementation of the pilot project was much less than the cost of the initial proposal for the development of a global EAI infrastructure. This is attributed to that the pilot EAI projects have a limited scope (e.g. to demonstrate the use of EAI in overcoming integration problems and not to integrate the whole organisation). In addition, the proposal for the pilot projects reduced risks, as these projects would be pilots, and not a permanent solution but be used to evaluate the wider impact of EAI. Consequently, the managing board accepted the new proposal and supported the implementation of the three application integration pilot projects.

[2] Strategic mode of integration requires organisations to make all the appropriate changes to their IT infrastructure and redesign all their business processes to support integration

In interpreting from the empirical data, it appears that EAI benefits and EAI technical capabilities were considered as factors that influence the adoption of EAI in OILCORP. The reason for this is that the organisation expected to evaluate EAI benefits (technical and business) before taking the decision for adopting application integration. In addition, costs and risks (barriers) were also considered as influencing factors to EAI adoption. For instance, when the cost of the project was dramatically reduced through the proposition of pilot projects, the managing board accepted its pilot adoption. Likewise, pilot projects will have less expected barriers and impact to OILCORP. For instance, resistance to change and politics issues are expected to be less comparing to a global EAI project. The reason for this is that the EAI solution will affect only a small number of employees and departments and only for a short period. Thus, their reaction will less since, the pilot EAI system will not be a permanent solution. For that reason, the organisation supported the implementation of these projects easily since it reduces risks and impact.

5.3.5 The IT Infrastructure

Each subsidiary has its own IT infrastructure, which causes a diversity of technical and organisational problems as reported in section 5.2.2. When the UK based EAI Project Manager (PM) was asked to comment on this situation he reported that:

> *"We knew we had to change things in order to improve our way of working and increase overall productivity. This situation with more than 1500 legacy systems and more than 90 ERP systems up and running has been an obstacle for our organisation... Then we had a lot of pressures to change our infrastructure. Increased competition pushes our management to find ways to gain competitive advantage through IT infrastructure... The managing board pushes us to find ways to reduce costs and increase flexibility. Our customers and suppliers pressurise us to work closer with them... We realised that to achieve this [integration] there is only one way and this is enterprise application integration."*

Interviewees were then asked to provide more details regarding the IT infrastructure of OILCORP. However, most of them were reluctant to answer due to confidentiality reasons.

Interviewees reported that it is OILCORP's belief that the IS that are used by the organisation form a type of competitive advantage. For that reason, interviewees could not provide detailed information. Nonetheless, general (not detailed) information regarding OILCORP's IT infrastructure was given and summarised below:

- **Custom systems:** The organisation consists of more than 1500 legacy and custom built applications (globally). Most of these systems run on mainframes environments and some of them are too old (they have existed since 70's). The number of legacy systems was greater a few years ago but, many of them were replaced when the organisation attempted to solve the year 2000 problem (Y2K). OILCORP collaborates close with IBM with a number of custom-built systems having been provided or supported by IBM. Most custom systems are complex and have a heterogeneous and incompatible nature. As a result, the integration of these systems has become a real obstacle to the organisation.

- **Packaged Applications:** Up to 90 enterprise resource planning systems exist in OILCORP and its subsidiaries around the globe. Most ERP systems were purchased from four major ERP vendors including SAP, JD Edwards and Oracle, with the majority of ERP systems purchased from SAP and JD Edwards. Many versions of these products exist in the organisation (e.g. SAP R/2, SAP R/3) running on a diversity of platforms (e.g. mainframes, windows NT).

- **E-business solutions:** During the last decade, the organisation has adopted many e-business enable applications including eProcurement, eSupply chain management, eCustomer relationship management, portal sites, eStores, EDI applications etc. The majority of these systems run over Internet oriented networks (e.g. Internet, intranets, extranets) with only few applications (mainly EDI based solutions) running over Value Added Networks (VAN). Most e-business solutions run over Windows or Unix operating systems.

It appears that the IT infrastructure at OILCORP influences the adoption of EAI. The limitations of the existing IT infrastructure and the problems that it causes the organisation stimulated the adoption of an integrated IT infrastructure. It is also clear (as stated by the P.M.) that pressures (e.g. management) at OILCORP have pushed the IT department to improve its infrastructure and finally led to the introduction of enterprise application integration. Thus, it appears that *pressures influences the adoption of EAI at OILCORP.* It seems that pressures came from both internal and external sources. External pressures appear to deal with competitors, customers and suppliers where internal pressures focus on factors such as managerial and technical issues. In many the cases, internal pressures derived from the limitations of the existing IT infrastructure and the need for increasing flexibility and productivity.

5.3.6 Evaluating Integration Technologies

Before proceeding to the implementation of the pilot projects, the IT department had to select *appropriate* EAI software for the development of these projects. Marketplace confusion around EAI products caused many problems to the selection of EAI software, as there is a plethora of EAI products promising integration (Ring and Ward-Dutton, 1999), with no single product addressing all integration types (e.g. data, component, custom, e-business applications etc). Although the EAI vendors promote their software packages as 'plug and play' solutions there is no EAI product offering an 'out of the box' automated integration. All EAI products require implementation and customisation, as each has its own advantages and drawbacks. In addition, many middleware vendors promote their software tools as EAI solutions (see Appendix B), which also cause confusion in the EAI marketplace. The reason for this is that a middleware product is often part of an EAI solution and not a solution itself. For all of these reasons, the IT department took a decision to evaluate EAI packages in-house, and to select the most appropriate. In doing so, a group of consultants studied and analysed the characteristics of the EAI products identified and set up a list of evaluation criteria. Then, a group of internal experts and consultants evaluated application integration solutions (EAI packages and integration technologies) according to these criteria.

When the interviewees asked to report the evaluation criteria they refused since OILCORP believes that such an evaluation framework is an important instrument for decision-making.

Since there remains a marketplace confusion regarding EAI, OILCORP can not share its evaluation framework with others (even academics) due to confidentiality reasons. Interviewees believe that sharing this framework with others may lead to possible loss of a competitive advantage. When the P.M. was asked to clarify this issue, he mentioned:

> *"... We invested money and time to come up with a comprehensive framework for evaluating EAI technologies. We did not have the knowledge as a department to develop this framework due to lack of understanding and lack of people with EAI skills. For that reason we hired consultants to do the job... We believe that other companies and our competitors will have problems too in evaluating EAI solutions due to market confusion. They will spend time to understand the market and take their decisions for adopting an EAI product. In our case, time is related to money and competitive advantage. Thus, this framework is a helpful tool for us and we do not want to share it with others."*

However, the interviewees informally reported a number of evaluation criteria. The list below is a subset of the criteria that were used for the evaluation and selection of EAI software. Among others the evaluation criteria included:

- *Total cost of ownership.* This cost includes the cost of purchasing an EAI package, the cost for annual licenses, development costs and maintenance;
- *Up front cost.* The start up cost should not be high. Thus, minimum changes and extensions of existing infrastructure (hardware cost) are required;
- *Global presence.* The vendor should have global presence or should have representatives around the world;
- *Flexibility.* The adopted EAI packaged should be able to respond to changes in the systems environment;
- *Maintainability.* The overall solution provided by the adoption of the package should allow changes without causing problems to other applications or systems;
- *Global scalability.* The package should provide scalability in integrating organisation's systems around the globe. This means that the adopted packaged should provide high

performance as greater demands are placed upon it, through the addition of extra computing power;

- *Product maturity.* The EAI software should be mature enough and should eliminate risks; and,
- *Integration capabilities.* The adopted EAI software should be appropriate for the integration of custom, packaged and e-business applications.

The last criterion presented (integration capabilities) was important for the justification of the EAI software selection, as OILCORP could not accept an EAI solution that presents difficulties in integrating the organisation's existing systems (ERP, custom, e-business). Before selecting the EAI software, the IT department took the decision to adopt SAP modules as the ERP system of the pilot projects. This decision was taken as the IT department tried to overcome the redundancy of ERP system functionality that OILCORP faced due to mergers and acquisitions. The reason for phasing out other ERP packages (e.g. JDE) and selecting SAP was justified, since the majority of ERP implementations in OILCORP was based around SAP modules. Furthermore, the organisation was satisfied by the functionality of SAP and prefers it to the other packages.

However, the decision made by the IT department predetermines the decision for the adoption of EAI software. It appears that the IT infrastructure (SAP package in this case) affected the decision for adopting appropriate EAI software. SAP has a close collaborator with an EAI vendor called CrossWorld whose software solutions achieve integration among SAP modules, e-business applications and legacy systems (Gilbert and Sweat, 1999). In addition, CrossWorld's software complied with most of the evaluation criteria set by OILCORP and it can collaborate with other software solutions to achieve process integration. Based on these, the expert group made the decision that CrossWorld's integration software could be adopted for development. Also, OILCORP adopted IBM's Message Queuing Series Integrator (MQSI) as the message brokering software. Tibco middleware and the Mercator's EAI solution were also adopted, since there is no single EAI package solving all integration problems. Mercator software and Tibco middleware were purchased to address legacy integration. A number of other software solutions were purchased by OILCORP but, they can not be reported due to confidentiality restrictions. All software solutions adopted can collaborate and provide a reliable platform for integrating OILCORP's applications.

In interpreting the empirical data, it appears that a framework for evaluating EAI technologies has a significant role in selecting integration software. As stated by the P.M., the development of a consistent evaluation framework can be considered as a strategic decision making tool that may result in achieving competitive advantage (by adopting appropriate set of EAI technologies). An evaluation framework for assessing EAI technologies seems to be a factor in influencing the adoption of enterprise application integration. The evaluation process followed by OILCORP has shown that no single EAI package addresses all integration problems. For that reason OILCORP took the decision to adopt various products to integrate its IT infrastructure. Another issue that came up from the evaluation framework is that OILCORP has used not only technical evaluation criteria (e.g. flexibility) but also non-technical (e.g. costs). Some of the latter criteria focus on costs and support, and which have been considered by the author as influencial factors for EAI adoption (see Chapter 3). Therefore, it appears that OILCORP's decision for adopting an EAI solution was influenced by factors that deal with the cost of an EAI package and vendor support.

5.3.6.1 Scope and Analysis of the Proposed Evaluation Framework for EAI Technologies

Three of the stakeholders being involved in the adoption and implementation of EAI at AUTOCORP were interviewed using structured interviews. Interviewees included: (a) an Internal Consultant (I.C.) who works at OILCORP UK; (b) an Integrator (Int.) who have involved in the EAI projects and he works at OILCORP Holland and, (c) the UK based EAI Project Manager (P.M.).

Initially, interviewees were asked to identify those requirements that an integration technology should fulfil to be adopted in an EAI project. Apart from the requirements that are summarised in Table 5.15 and refer to the four categories of the evaluation criteria, interviewees were encouraged to report additional requirements.

Integration Requirements - Evaluation Criteria	I.C.	Int.	P.M.
Maintainability	●	●	●
Flexibility	●	●	●
Scalability	●	●	●
Portability	●	●	●
Reusability	●	●	●
Product Maturity	–	◒	◒
Technology Maturity	–	◒	◒
Complexity	–	●	◒
Non-invasive	●	●	●
Performance	◒	●	●
Real Time Integration	●	●	●
Mainframe Compatibility	●	●	●
Non-Mainframe Compatibility	●	●	●
Support of Data Integration	●	●	●
Support of Objects Integration	●	●	●
Support of Process Integration	●	●	●
Support of Transportation Layer	●	●	●
Support of Transformation Layer	●	●	●
Support of Process Automation Layer	●	●	●
Support of Custom-to-Custom Application Integration	–	●	◒
Support of Custom-to-Package Application Integration	●	●	●
Support of Custom-to-e-business Application Integration	●	●	●
Support of Packaged-to- Packaged Application Integration	●	●	●
Support of Packaged –to-e-business Application Integration	●	●	●
Support of e-business-to- e-business Application Integration	●	●	◒
Support of Custom-to-Packaged-to- e-business Application Integration	●	●	●
Other: Security	●	●	●
Other: Manageability	–	●	●
Other: Vendor's Global Presence	–	–	●
Other: Vendor's Support	●	●	●
Other: Cost	●	–	●

Table 5.15: Identification of Integration Requirements at OILCORP

As illustrated in Table 5.15, there appears to be a similarity in interviewees' perceptions. These perceptions deal with the integration requirements that should be considered when evaluating integration technologies. However, there is a difference in their answers regarding: (a) product and technology maturity; (b) product complexity; (c) custom-to-custom application integration and, (d) e-business-to-e-business integration. Internal consultant did not report a level of importance for the aforementioned requirements as he was quite confused regarding these issues. The project manager believes that product complexity, custom-to-custom application integration and e-business-to-e-business integration are of medium importance and he states that:

> *"The complexity of a technology is of medium importance as there is a lack of technologies that solve integration problems. Once you find out that a technology addresses most of your integration problems you are happy to use it. This means that the issue of complexity is not so important... Our intention is to integrate all applications and not only custom-to-custom or e-business-to-e-business systems. Indeed, we need technologies to support these types of integration but this is not of high importance."*

The integrator reported high level of importance for these requirements based on the nature of his work, and says that:

> *"You always run out of time when you integrate your systems due to the overall complexity. As management pushes you to keep milestones, you do not have time to use technologies with high complexity. You only use complicated technologies when there is no other technology available... For me it is important to have tools that support my job. So, I need tools that support not only custom-to-packaged-to-e-business integration but all types of integration [e.g. custom-to-custom] since I integrate these types of systems."*

Interviewees identify more evaluation criteria including: (a) security; (b) manageability; (c) vendor global presence; (d) vendor support and, (e) cost. As mentioned by all interviewees, security is of high importance when integrating IT infrastructures. Security issues like: (a) which users have access to a specific process or data set and, (b) what level of access do suppliers or customers have on the integrated IT infrastructure should be considered. In addition there is a need for integration technologies with security features. These findings are along the same lines as Ring and Ward-Dutton (1999) who suggest that manageability and security are significant characteristics of an integrated IT infrastructure.

It appears that the job description influences the perception of interviewees regarding the significance of specific integration requirements/criteria. Moreover, there exist differences in the answers of interviewees regarding additional criteria such as cost, vendors' support, global presence etc. It seems that integrator does not consider the cost of a technology as an integration requirement. The reason for this is that integrator's tasks deal with the development of an integrated system and not with the cost. In contrast, the project manager reports the cost of a technology as a requirement of high importance. The reason for this is that P.M. has a fixed budget for the project and can not exceed it (in case the cost of a technology is high).

It seems that requirements like vendors' support, global presence and costs are related with non-technical factors (EAI cost, support) that influence the adoption of a technology. It also appears that costs and support are considered as influential factors for EAI adoption. Interviewees were then asked to assess the integration technologies according to the four categories of evaluation criteria identified in section 3.4. The purpose of the evaluation is to investigate whether there is a single integration technology that meets all evaluation criteria and thus, addressing all integration problems. Moreover, the evaluation seeks to examine the proposed evaluation framework in terms of whether: (a) it supports decision-making; (b) improves IT sophistication and, (c) influences the adoption of appropriate integration technologies.

Table 5.16 presents the assessment for integration technologies using applications' elements (data, objects, processes) as evaluation criteria.

	Applications Elements								
	Data			Objects			Process		
Integration Technologies	**I.C.**	**Int.**	**P.M.**	**I.C**	**Int.**	**P.M.**	**I.C**	**Int.**	**P.M.**
ODBC	●	●	●	–	○	–	–	○	–
JDBC	●	●	●	●	●	●	–	○	–
RPC	–	●	●	–	–	–	–	–	–
MOM	●	●	●	○	○	○	○	○	○
Message Broker	●	●	●	○	◒	○	●	●	●
XML	●	●	●	●	●	●	○	◒	○
TPM	○	◒	◒	○	○	○	●	◒	◒
Application Serves	●	●	●	●	●	◒	○	◒	○
CORBA	●	●	●	●	●	●	◒	○	○
DCOM / COM	◒	◒	◒	●	●	◒	◒	○	○
EJB	○	○	○	◒	◒	●	◒	○	○
Screen wrapper	◒	◒	○	○	◒	○	○	○	○
APIs	◒	◒	◒	●	●	●	○	○	○
Adapters	●	●	●	●	●	●	○	○	○

Table 5.16: OILCORP-Evaluating EAI Technologies Using Applications Elements as Criteria

It appears that there is no single technology supporting all applications' elements. Even if a technology could integrate all applications' elements, this would not solve the integration problems with the integrator reporting that:

> *"It seems that we need a single technology that achieves data, objects and process integration since these [data, objects] are the elements that exchanged among application; but it is not like this. What we need are tools to support the integration of various systems. Each system requires a tool for extracting and inputting applications' elements... Because, most applications were developed using heterogeneous programming languages and technologies, a set of tools is required to support all systems. Some of these tools will address data integration where others objects, components etc. For us, the extraction and insertion of data is like a separate integration layer that deals with the connectivity. Through this layer you gain access to source and target applications and become capable of retrieving or inputting data... For us it is important to have tools that support the connectivity [inputting and extracting*

data to/from applications], the transportation, the translation and the process automation layers."

This perspective is different from what most of the literature findings reported in Chapter 3 (Duke *et al.*, 1999; Linthicum, 2000a; Ruh *et al.*, 2000) as these findings do not consider connectivity as an integration layer. Nonetheless, the perspective of integrator is similar to Ring and Ward-Dutton (1999) approach who consider connectivity as integration sub-layer. Ring and Ward-Dutton (1999) suggest that connectivity is part of transportation layer. Integrator's perspective appears to be more realistic. The reason for this is that integration is not only about transferring and translating information among applications but also, connecting (linking) applications and integrating business processes. Thus, the author now considers connectivity as an integration layer and classifies integration technologies based on the aforementioned four layers namely: (a) connectivity; (b) transportation; (c) transformation and, (d) process automation. Table 5.17 illustrates that integration technologies may support one or more layers.

Integration Layer	Integration Technology	Reference
Process Automation	Message broker	Linthicum (2000a)
Transformation	Message broker	Linthicum (1999)
	Adapters	Ring and Ward-Dutton (1999)
	XML	Cingil *et al.*(2000)
	CORBA	Mowbray and Zahavi (1995)
Transportation	RPC	Ruh *et al.*(2000)
	MOM	Ring and Ward-Dutton (1999)
	XML	Usdin and Graham (1998)
	TP Monitors	Serain (1999)
	Application Servers	Ruh *et al.*(2000)
	CORBA	Zahavi (1999)
	COM/DCOM	Andrew (1998)
	EJB	Wutka (2001)
Connectivity	ODBC	Serain (1999)
	JDBC	Ring and Ward-Dutton (1999)
	XML	Morgenthal and La Forge (2000)
	CORBA	Wutka (2001)
	COM/DCOM	Rosen (1998)
	EJB	Wutka (2001)
	Screen Wrappers	Serain (1999)
	APIs	Linthicum (2000a)

Table 5.17 Classification of Integration Technologies Using Integration Layers

Although technologies like ODBC, JDBC, screen wrappers and APIs support data transformation, the author classifies them as technologies that support more the connectivity layer rather than the translation layer. The reason for this is that in most cases these technologies are used at application level (e.g. to extract data from source application) rather than at integration level (e.g. as part of an integrated infrastructure [e.g. hub and spoke integration architecture]). This is also supported by Ring and Ward-Dutton (1999) and Ruh *et al.*(2000), as well as by interviewees' in this case study.

Table 5.18 presents interviewees' evaluation results when they assess the integration technologies using the integration layers as evaluation criteria. This set of criteria comprises part of the framework analysed in Chapter 3. Based on the interviewees' answers it appears that message brokers is the only technology that supports the transportation, transformation and process automation layers. However, based on Table 5.17, message brokers do not support the connectivity layer (which was introduced during the analysis of the case studies [see previous paragraphs]). As a result, it appears that none of the existing technologies supports all four integration layers (connectivity, transportation, transformation and process automation). Interviewees report that message brokers are often used as the core integration mechanism, with developers using their functionality at the highest layers of integration (transformation, process automation). This finding is also supported by literature (Duke *et al.*, 1999; Edwards and Newing, 2000) with Linthicum (2000a) suggesting that message brokers are the *main engine* of an integrated solution, and support the coordination of integration technologies and tasks. Based on interviewees' perceptions, it appears that message brokers in practice are used to support process the automation and the translation layer (although they can facilitate transportation layer).

Table 5.19 summarises the interviewees' evaluation results when they assessed various integration technologies while using the seven system types (e.g. custom-to-custom). This set of evaluation criteria also comprises part of the evaluation framework presented in Chapter 3.

	Integration Layers								
	Transportation			Transformation			Process Automation		
Integration Technologies	I.C.	Int.	P.M.	I.C	Int.	P.M.	I.C	Int.	P.M.
ODBC	○	○	○	◒	●	◒	-	-	-
JDBC	○	○	○	◒	●	◒	○	○	○
RPC	●	●	◒	○	○	○	-	-	-
MOM	●	●	●	○	○	○	◒	○	-
Message Broker	●	●	●	◒	●	●	●	●	●
XML	●	●	●	●	●	●	○	○	○
TPM	◒	◒	○	○	○	○	◒	○	-
Application Serves	◒	◒	◒	○	○	-	●	○	-
CORBA	◒	◒	○	◒	◒	◒	◒	◒	○
DCOM / COM	◒	◒	◒	◒	○	○	◒	◒	○
EJB	◒	◒	◒	○	○	○	-	○	-
Screen wrapper	○	○	○	●	●	●	○	○	○
APIs	○	○	○	●	◒	◒	○	○	○
Adapters	○	○	○	●	●	●	○	○	○

Table 5.18: OILCORP-Evaluating EAI Technologies Using Integration Layers as Criteria

	System Types																				
	Custom-to-Custom			Custom-to-Packaged			Custom-to-e-business			Packaged-to-packaged			Packaged-to-e-business			E-business-to-e-business			Custom-to-Packaged-to-e-business		
Integration Technologies	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.
ODBC	○	◒	○	◒	◒	◒	◒	◒	◒	●	●	●	●	●	●	●	●	●	●	●	●
JDBC	○	○	○	◒	◒	◒	◒	◒	◒	◒	◒	◒	●	●	●	●	●	●	◒	◒	◒
RPC	○	○	○	◒	◒	◒	○	◒	○	○	○	○	○	○	○	○	○	○	○	○	○
MOM	◒	◒	◒	◒	◒	◒	◒	◒	○	◒	◒	◒	◒	◒	◒	○	○	○	◒	◒	○
Message Broker	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
XML	○	◒	○	◒	●	◒	●	●	●	●	●	●	●	●	●	●	●	●	◒	●	●
TPM	○	◒	◒	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Application Serves	–	○	○	–	◒	◒	◒	●	◒	◒	◒	◒	●	●	●	●	●	●	◒	◒	◒
CORBA	–	◒	◒	–	◒	◒	○	◒	◒	◒	●	◒	○	●	●	○	●	●	◒	◒	◒
DCOM / COM	○	○	○	○	◒	○	–	◒	◒	◒	●	◒	○	●	◒	○	●	◒	◒	●	◒
EJB	○	○	○	○	○	○	◒	◒	◒	◒	◒	●	○	◒	●	○	●	●	◒	◒	◒
Screen wrapper	●	●	●	◒	●	●	●	●	●	○	○	○	○	○	○	○	○	○	◒	◒	◒
APIs	○	○	○	◒	◒	◒	○	○	○	●	●	●	○	●	●	○	◒	◒	◒	◒	◒
Adapters	◒	◒	◒	●	●	●	◒	◒	◒	●	●	●	●	●	●	○	◒	◒	●	●	●

Table 5.19: OILCORP-Evaluating EAI Technologies Using System Types as Criteria

As presented in Table 5.19, interviewees share similar perceptions regarding the capability of integration technologies to piece together system types. It appears that message brokers support the integration of all system types. This finding is in accordance with Linthicum (2000a), but it does not recommend that message brokers can be used to solve all integration problems. The reason for this is that message brokers present other limitations (e.g. do not efficiently support objects integration) and therefore, they do not overcome all integration problems. OILCORP considered among others four categories of evaluation criteria (applications' elements, integration layers, system types and integration requirements) before taking a decision for adopting an integration technology. This perspective is supported by the P.M. who reported that:

> *"Applications' integration is a complicated problem. We have to overcome problems at various levels like application elements, integration layers and systems that are integrated. It is not enough to select a technology that addresses part of the problems. For that reason we have to purchase technologies that address multiple integration problems and meet a variety of evaluation criteria."*

In addition to message brokers, interviewees emphasised on nature of adapters and made useful comments. Based on Table 5.19, it appears that adapters result in a medium to high level of integration, when piecing together system types. Nonetheless, this is not a fixed rule but represents possible solutions that, adapters could offer. Interviewees mentioned that each EAI vendor implements its adapters by focusing on specific system types. For instance, Mercator's adapters (that are included in Mercator's EAI software) focus more on custom systems integration rather than on packaged applications integration. Recently, Mercator developed adapters that achieve packaged application integration but Mercator's adapters support only SAP R/3 ERP systems (Linthicum, 2000b; 2001) and not other ERP or packaged system. For that reason, the integrator suggested that it is important to understand the level of applicability for each adapter when selecting one (adapter). For that reason, the proposed evaluation framework may need modifications to address this issue, since it will allow organisations to realise the full range of functionality that a technology supports. However, the range of functionality appears to be different from EAI product to EAI product, as each vendor develops its products independently by focusing on specific market needs.

Another important issue that came from this discussion is that *EAI packages consist of a set of integration technologies that cover part or all integration layers.* EAI vendors support integration layers by configuring a package that consists of various integration technologies (products). EAI packages may be based on a mixture of vendors' products or third-party products. For instance, BEA's elink EAI package consists of BEA's products that cover the connectivity, transportation and process automation layers, and collaborates with Mercator message broker at a translation layer. *As a result, it seems that an evaluation framework that supports the assessment of integration technologies is not adequate to select appropriate EAI packages. In depth knowledge of each EAI package (and the integration technologies that it uses) is needed to understand the applicability of the solution that each package supports.* This situation causes additional confusion regarding the adoption of EAI packages and therefore, an extension of the proposed framework is required for evaluating and adopting not only integration technologies but also EAI packages.

Tables 5.20 and 5.21 present the evaluation results when interviewees used the integration requirements as evaluation criteria. Clearly, Tables 5.20 and 5.21 show that there is no single integration technology that meets all integration requirements. However, it seems that message brokers meets nearly all integration requirements. In addition, adapters and XML support most of criteria. This may provide an explanation why XML, adapters and message brokers have a significant role in EAI solutions and packages.

	Integration Requirements																	
	Maintainability			Flexibility			Scalability			Portability			Reusability			Maturity		
Integration Technologies	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P M.	I.C.	Int.	P.M.
ODBC	○	○	○	◐	○	○	○	◐	◐	◐	●	◐	◐	◐	◐	●	●	●
JDBC	●	◐	◐	◐	◐	◐	○	○	○	●	◐	◐	◐	●	◐	◐	◐	◐
RPC	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●
MOM	◐	◐	○	◐	◐	◐	◐	◐	○	○	○	○	○	○	○	●	●	●
Message Broker	●	●	●	●	●	●	●	●	●	◐	●	●	●	●	●	○	◐	◐
XML	●	●	●	●	●	●	○	◐	◐	●	●	●	●	◐	●	◐	◐	◐
TPM	–	○	–	○	○	○	●	●	●	●	●	●	–	–	–	◐	◐	◐
Application Serves	◐	◐	◐	◐	◐	●	●	◐	●	●	●	●	◐	●	●	◐	◐	◐
CORBA	◐	◐	◐	◐	◐	◐	◐	◐	◐	○	◐	○	●	●	◐	◐	◐	◐
DCOM / COM	◐	◐	◐	◐	◐	◐	–	◐	–	○	○	○	●	●	●	◐	●	●
EJB	●	◐	●	◐	◐	●	○	◐	–	○	◐	◐	●	●	●	○	○	○
Screen wrapper	○	○	○	○	○	○	◐	◐	◐	–	○	–	○	○	○	●	●	●
APIs	◐	◐	◐	○	○	○	○	○	○	○	○	○	◐	◐	◐	●	●	●
Adapters	●	●	●	●	●	●	●	●	●	●	●	●	◐	●	◐	◐	◐	◐

Table 5.20: OILCORP-Evaluating EAI Technologies Using Integration Requirements as Criteria

Integration Technologies	Integration Requirements: Complexity			Non-invasive			Performance			Real-Time			Mainframe Compatible			Non-Mainframe Compatible		
	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.	I.C.	Int.	P.M.
ODBC	○	○	○	◐	◐	○	●	●	●	○	○	○	○	○	○	●	●	●
JDBC	○	○	○	◐	◐	◐	●	●	◐	○	◐	○	–	○	–	●	●	●
RPC	●	●	●	○	○	○	○	○	○	●	●	●	◐	●	●	●	●	◐
MOM	●	●	●	○	○	○	◐	◐	○	○	–	○	●	●	●	●	●	●
Message Broker	○	○	○	●	●	●	●	●	●	◐	◐	◐	◐	●	●	●	●	●
XML	○	○	○	◐	◐	◐	◐	◐	◐	◐	◐	○	–	○	○	●	●	●
TPM	●	●	●	○	○	○	◐	●	●	–	–	○	◐	◐	○	◐	◐	●
Application Serves	◐	◐	◐	○	○	○	○	◐	◐	◐	◐	○	◐	◐	○	●	●	◐
CORBA	◐	◐	◐	○	○	○	◐	◐	◐	●	◐	◐	●	●	●	●	●	●
DCOM / COM	◐	○	○	○	○	○	◐	◐	◐	●	●	◐	◐	○	○	●	●	●
EJB	●	◐	◐	○	○	○	○	○	○	◐	●	●	○	○	○	●	●	●
Screen wrapper	●	●	●	●	●	●	○	○	○	◐	○	◐	●	●	●	◐	◐	◐
APIs	●	●	●	○	◐	◐	○	◐	◐	●	●	●	◐	◐	◐	●	●	●
Adapters	○	○	○	●	●	●	◐	◐	◐	◐	●	◐	●	●	●	●	●	●

Table 5.21: OILCORP-Evaluating EAI Technologies Using Integration Requirements as Criteria

Regarding the proposed evaluation framework, there was a great deal of emphasis expressed by all interviewees on its importance to support decision-making, and improve IT sophistication. This was emphasised by the integrator, who said:

> *"Before adopting integrated solutions we need to understand all these technologies and find out which of them suits our infrastructure... Since there was a lack of integration skills in our group, a framework like this helps our efforts. With such an evaluation framework we can understand and compare all technologies. This improves our level of understanding and supports decision-making. However, the framework needs further criteria [e.g. security] to support the evaluation of EAI packages."*

Then, interviewees were asked whether an evaluation framework like the proposed one influences the adoption of EAI. As reported by interviewees the use of a similar framework (by OILCORP) has influenced the decision for EAI adoption. This was addressed with comments from project manager, who said:

> *"Sufficient knowledge of the integration marketplace allows us to make decisions regarding EAI adoption. If existing EAI packages and technologies had not satisfied us we would not have adopted them. So it is important to understand and evaluate EAI before adopting a solution."*

Based on these comments, it can be pointed out that an evaluation framework that assesses the capabilities of EAI packages and technologies can be viewed as an influential factor for EAI adoption. After choosing appropriate EAI packages and integration solutions, OILCORP has moved to the implementation of three pilot projects.

5.3.7 Pilot Projects

Interviewees reported that there is a lack of business understanding regarding what IT can provide and what are the differences among EAI and point-to-point connectivity. In support of this, the internal consultant mentioned that:

> *"Organisations are not convinced why they need EAI. They believe that they can solve their problems by using point-to-point interconnections."*

For that reason, OILCORP took the decision to implement pilot EAI projects to demonstrate the benefits and impact (barriers) of this technology on the organisation. Thus, it appears that the parameters that were used by OILCORP to evaluate the pilot EAI projects (e.g. benefits, barriers, costs, etc.) were factors that influenced the adoption of EAI in this organisation.

The existing IT infrastructure of OILCORP consists of a variety of systems such as ERP, custom applications (e.g. legacy systems, databases, data-warehouses) and e-business solutions (e.g. e-Procurement). The pilot projects were designed to address the diversity of system types that existed in OILCORP based on three different scenario-projects. In this section, pilot EAI projects at OILCORP are presented and then analysed.

5.3.7.1 Pilot Project One

The first pilot project started in January 2000 and finished 9 months later (September 2000). The aim of this project was to prove that application integration can be used to piece together various types of applications, such as legacy systems, e-business solutions, databases and ERP applications. Initially, pilot system analysis and design took place with the IT department spending much more time for these phases than in traditional system design. More specifically, the IT department spent 60% of their total pilot implementation time on system design and 40% on system development. The reason for this was that application integration required business process reengineering, which takes a longer time, as a number of systems have to be changed or phased out to support an integrated process. The more systems collaborate to automate a process the more difficult is to redesign that process.

The development of all pilot system was based on a process driven integration. This means that developers firstly integrated business processes and then systems. In doing so, they started from a system that triggers a business process and then integrate them with a central integration infrastructure. Then, they moved to the next application that automates the second part of the same process and pieced them together with the integration infrastructure.

Developers repeated this task sequentially until they achieved the integration of the whole business process. In doing so, all IS that automate parts of the same process were incorporated with the central integration infrastructure.

The integration infrastructure was based on a hub and spoke integration mechanism. In such a mechanism, applications are connected to a central hub (message broker) that contains the rules for connecting application together. A message broker was based on the hub and spoke mechanism and collaborated with other technologies to move data from one application to the hub and from the latter to the target application. The hub audits and coordinates the integration task and triggers relevant events. The message broker translates applications' elements by collaborating with other technologies (e.g. adapters, XML). Crossworlds message broker supports synchronous communication, which is important for real-time applications (see Appendix B). Also, the message exchange was based on an asynchronous communication model using other products (e.g. IBM MQSeries), which enables applications to operate independently without forcing source applications to wait to receive the results of their requests. Message brokers integrate multiple business process and applications through supporting data and message transformation, message filtering and routing. In addition, they provide rule processing capabilities, hosting business functions, message translation engines and bridges to many different platforms and applications (by using pre-built EAI adapters or existing APIs). The adoption of Crossworlds EAI software with MQSeries indicates that the former as an EAI package can not address all integration requirements.

Initially, OILCORP started integrating a business process that is related to e-supply chain management. In doing so, developers firstly incorporate the appropriate e-business modules using the hub and spoke integration mechanism.

During this stage, data from e-business applications were formatted based on HTML and XML standards and send to a web server. The web server sends the messages to the hub (message broker), which reformats the data based on SAP's data or objects structure. Then the data are sent to SAP modules and processed by the system. The results are then passed to e-business applications through the hub.

A legacy application was then incorporated with the systems (e-business applications – SAP) through the central hub. In many legacy systems, the user interface is the only available mechanism to access data, logic and processes. A category of tools called screen wrappers were then used to reuse useful business rules and data from existing applications and, (or) to ensure integrity between new and existing systems. Screen wrappers encapsulate (extract) data from user interfaces and transform them into raw data (screens as data) or objects (screens as objects). In doing so, they use mapping techniques to map the user's interface information to raw data or objects. In the case of OILCORP, legacy data were mapped and transformed into data or objects, sent to the central hub and then forwarded to either e-business applications or SAP.

Thereafter, a database was integrated with the other integrated systems (e-business, SAP and legacy). In support of this, interface drivers (e.g. ODBC) were used to simplify database access. When an application needs data from the database, it sends a request to a central hub. This hub retrieves data using interface drivers and then reformats data into the target's application format and transmits them to a target application. Likewise, an application may update one or more database Tables with new data records. Figure 5.10 depicts the integration of the aforementioned systems (e-business, SAP, legacy and database).

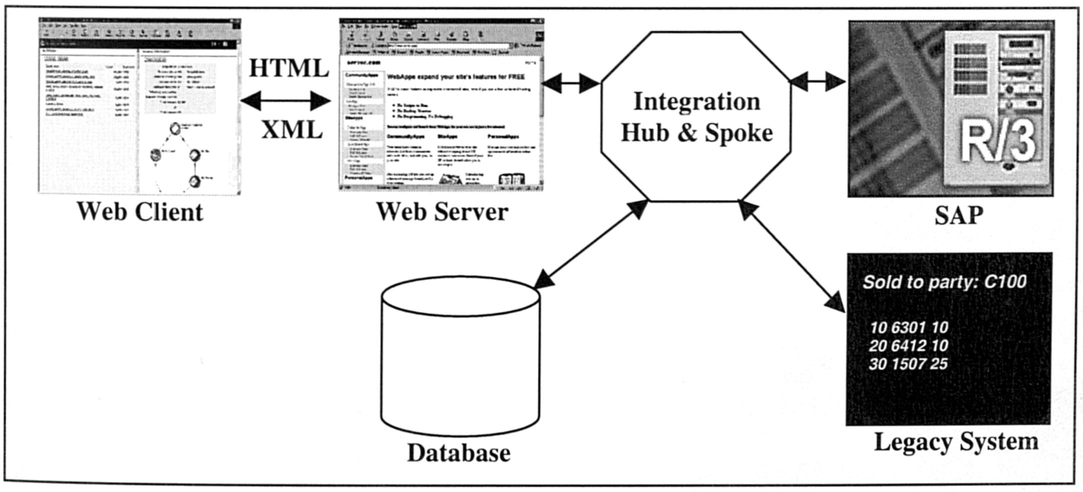

Figure 5.10: Pilot Project One - Integration of Custom, Packaged and e-business Systems

The implementation of the first EAI pilot project provided advanced capabilities to the IT department and increased the functionality and efficiency of the information systems. A number of business processes were redesigned, automated and integrated by the pilot system. Important benefits were derived from this pilot project including among others: (a) quicker response in change; (b) achieves customer satisfaction; (c) improves data quality; (d) achieves process integration and, (e) improves management and supports decision-making. Benefits that are elicited from pilot EAI projects are analysed in section 5.3.9.

The IT department used this pilot system to run various business cases and demonstrate the systems functionality and benefits. Among others, the integrated pilot system is capable of running many scenarios including the following case as described by the internal consultant:

> *"A customer accesses a web application through the Internet to place an order. Before confirming the order, the web application should check both stock availability and customer record. The web application contacts the hub and makes a request for the information needed. The hub retrieves the data from both the database that handles customers' records and the legacy system that deals with stock availability. Then it sends data to the web application. If there is no problem with the order, the web application updates the customer record in the database and the stock availability at legacy system. In addition, it sends all financial details to SAP's financial module and order details to SAP orders module."*

The functionality of this EAI pilot project and the benefits that offered to OILCORP led the organisation to the conclusion that it was a successful project. Therefore, after the end of this pilot project the managing board started changing its stance regarding the adoption of a global EAI IT infrastructure.

5.3.7.2 Pilot Project Two

OILCORP has established a global trading network, which consists of a number of separate companies. While remaining distinct entities, responsible for running their own business, these companies use the OILCORP's trading identity to demonstrate their close working relationships and present a global service to their customers. The network companies provide a trading interface with customers and the market. The aim of all OILCORP trading companies is to provide flexibility, rapid response, short lines of communication and service oriented to the demands of customers throughout the world.

The second pilot project was implemented in one of these OILCORP companies dealing with trading and shipping. In the context of this dissertation, the author uses the name OILCORP SUB_A to refer to this company due to confidentiality reasons. OILCORP SUB_A trades nearly five million barrels of crude oil and products, and moves cargoes on 140 deep-sea tankers and gas carriers around the world on a daily basis. OILCORP SUB_A shipping division runs a fleet of 53 ocean going vessels, both owned and demised or managed, comprising 32 oil tankers and 21 gas carriers, and a further 13 vessels on time charter.

At a managerial and technical level, OILCORP SUB_A has realised that the existing IT infrastructure has caused many problems in implementing its business strategy. An internal consultant reported that OILCORP SUB_A consists of a diversity of incompatible systems that handle important information. For instance, there exist multiple systems dealing with customer data. The heterogeneity of existing systems had led the company to develop multiple datasets for the same entity. For instance, an application running on a legacy system has its own Table for a specific customer (e.g. CUST_CORP_A). Likewise, other applications (e.g. an SAP R/3 system, and an e-business module) have their own tables/objects for the same customer. These limitations have therefore caused many problems to the company such as:

- Lack of compatibility since, each system follows its own structure and standards.

- Data quality. There is a confusion regarding the more updated data (and their location). Many datasets include some data that are the more recent but they also include other data that are not updated. For instance, a dataset may have updated data dealing with the order details of CUST_CORP_A. However, the same dataset may consist of data that are old such as the customer details (e.g. CUST_CORP_A contact details).

- Co-ordination among applications. The heterogeneity and incompatibility among information systems has resulted in bad co-ordination. These problems have also an impact on the efficiency and productivity of the OILCORP SUB_A, since there has been a redundancy of data and tasks due to systems incompatibility. The explanations for these problems are similar to those reported earlier in this case study (see section 5.3.2, 5.3.3).

- Decision-making. The aforementioned problems regarding: (a) data incompatibility; (b) data quality and, (c) the co-ordination between applications affect the decision making process. The reason for this is that the management can not effectively use the data and its IT infrastructure to support decision making. Thus, many decisions are not accurate which lead to managerial problems as it has explained in earlier sections of this chapter.

As reported by an internal consultant, OILCORP SUB_A has to change its IT infrastructure *to better implement its strategy* (e.g. *do business more effectively*) and meet the aim of OILCORP trading network. For that reason, there is a need for:

- A single data source that consists of the more recent data. Such a data source will support decision making and data quality;
- An integrated IT infrastructure, that address the problems reported earlier in this section. This will result in the development of a flexible and manageable infrastructure; It will also support a high quality and cost effective, commercial services that are based on a common flexible IT infrastructure; and,
- An IT infrastructure that will allow the company to run e-business more effectively and take advantage of digital economy.

In this context, the second EAI pilot project has attempted to address part of this need. Its aim was to synchronise customer data and create a single customer entity. In doing so, developers incorporated applications that handle customers' data (such as packaged systems and legacy applications) with a master data reference model. The latter was developed for the purposes of this project and it functions like a data warehouse. The master data reference model collects data from other applications (e.g. an SAP R/3) and stores the more recent customer data. Thus, the synchronisation of customer data is easier since there is a single customer entity. This results in data quality and allows management to take more accurate decisions.

EAI pilot project two started in December 2000 and finished at the end of February 2001. Along similar lines with the first pilot project, the implementation was based on process driven integration. Data are extracted from an SAP R/3 system as well as from custom-built applications dealing with clearance and trading. The master data reference model has an enhanced functionality since it follows much of the concepts of a message broker (see Appendix B). Thus, it uses business rules and logic to integrate, automate and synchronise data and processes. Mercator EAI software, IBM MQ Series, TIBCO middleware and other products were used for the implementation of this project. OILCORP SUB_A used Mercator EAI package since it supports customs application integration and provides adapters for SAP R/3 modules. IBM MQ Series was used for data exchange and for asynchronous communication between applications. Tibco middleware supported the integration of custom applications. Figure 5.11 illustrates the second EAI pilot project.

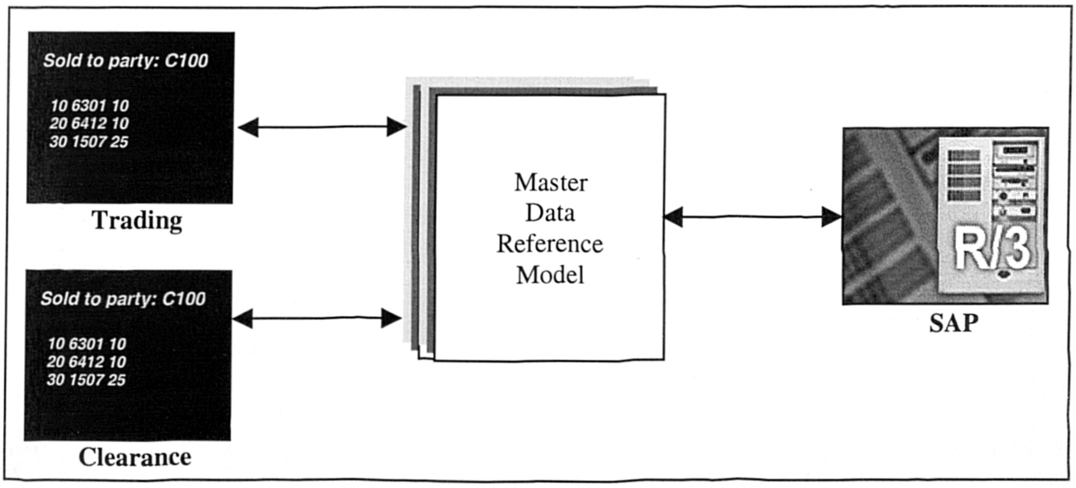

Figure 5.11: Pilot EAI Project Two

This EAI pilot project achieved its goal for developing a single data source for customer data that supports synchronisation. It appears that the use of master data reference model that is based on the functionality of a message broker improves the overall performance of this data reference model. Based on business logic and business rules the integration and synchronisation of customer data and processes can be achieved in more sophisticated and effective way. Thus, it increases the performance, reliability and integrity of the overall solution. At this end, an integrator reported, that the use of EAI technologies (e.g. Mercator software) has resulted in the development of a flexible and manageable master data reference model. It therefore, appears that the use of EAI technologies has successfully resulted in: (a) the development of a single customer data entity and, (b) the synchronisation of customer data. These has also resulted in better decision-making regarding customers and improved the efficiency of OILCORP SUB_A.

5.3.7.3 Pilot Project Three

The third EAI pilot project started in March 2001 and it is a two-year pilot project. The project is being implemented by a subsidiary of OILCORP that focuses on European oil products. Similarly to the previous cases, the author uses the name OILCORP SUB_B to refer to this company.

During the recent years, OILCORP SUB_B has faced pressures from its trading partners as well as from competitors. OILCORP's SUB_B customers and suppliers have demanded a closer collaboration with the company. However, OILCORP SUB_B could not respond to its trading partners demands due to organisational and technical problems. Its existing IT infrastructure presents various limitations similar to those reported in earlier sections (see section 5.3.7.1, 5.3.7.2). The company consists of a number of incompatible IS that include: (a) a diversity of ERP systems (e.g. J. D. Edwards, Oracle, SAP); (b) custom built systems (e.g. EDI applications, manufacturing systems) and, (c) e-business solutions (e.g. eProcurement, eSales). As reported by a manager, the diversity and the heterogeneity of these systems was an obstacle for achieving closer collaboration with trading partners. Thus, OILCORP SUB_B has to overcome many technical problems such as:

- data quality;
- systems redundancy;
- difficulties in marinating the existing IT infrastructure and,
- problems in migrating ERP modules to SAP R/3.

A closer collaboration with trading partners requires changes not only to the IT infrastructure but also to the organisational structure. When an internal consultant was asked to comment he said:

> *"We know that the existing IT infrastructure is an obstacle for creating tighter links with our trading partners... The company believes that the development of an integrated IT infrastructure will support a more functional and operable infrastructure that will efficiently support closer collaboration with our customers and suppliers. Such an infrastructure will reduce the redundancy of systems and result in data quality and process improvement. This will also require changes in our structure, business processes and practices. Thus, we have to change our structure in order to adapt a closer model of collaboration."*

As a result, OILCORP SUB_B has recently started transforming its business operations to become more 'customer-focused', according to a company press release. Thus, the company attempts to align the manufacturing, supply and distribution activities in Europe into one separate entity to optimise profits across the total Europe supply chain. Beneath this single European unit, the manufacturing, supply and distribution activities will be grouped into three broad geographic areas.

In implementing this policy, OILCORP SUB_B aims to provide increased service efficiencies to its customers by consolidating transaction systems onto a single platform. The new infrastructure will be key to the company's continuing success and will mean it can respond more quickly to customer demand: enhancements to existing products and services can be easily introduced and OILCORP SUB_B will be able to expand into new product areas. It will also reduce duplication of effort across different countries.

The third EAI pilot project, aims at investigating the use of EAI technologies in OILCORP SUB_B. In doing so, the company took the decision to run a pilot project that focuses on the development of an integrated global product catalogue. This project was based on a similar scenario to the pilot project two. The purpose of pilot project three was to develop a master data reference model that deals with a single global product catalogue.

As illustrated in Figure 5.12 data are extracted from SAP R/3 and legacy systems (e.g. global product catalogue) and sent to the master data reference model. Data are extracted using EAI packages such as CrossWorlds and Mercator. OILCORP provided OILCORP SUB_B with EAI packages therefore OILCORP SUB_B did not evaluate EAI packages. This company has used two different EAI packages since: (a) Mercator provides a wide range of adapters that support custom built applications integration. Mercator software supports only loose type of integration which is more appropriate for asynchronous communication and, (b) CrossWorlds software focuses more on packaged applications integration and supports both loose and tight integration. IBM MQ Series have been used to support the transportation layer. In doing so, they transfer data from ERP systems and the legacy application to the master data reference model. The connectivity layer is supported by: (a) APIs in the case of ERP packages and, (b) screen wrappers in the case of legacy system.

The master data reference model uses business logic and rules to integrate and automate data that are extracted by the source applications (ERP systems and global product catalogue). The master data reference model stores unique records for each product. In doing so, it achieves data quality and data homogeneity. Additionally, it provides a standard interface to other applications that support decision-making. The reason for this is that the management can easily better retrieve and analyse data since there is a single source of updated data.

As mentioned above the third EAI pilot project will be finish in March 2003. For that reason, there are no available data regarding the success of the project. However, the author interprets empirical data from this pilot dealing with the adoption of EAI. As a result, it can be summarised that:

- External pressures such as: (a) trading partners pressures and, (b) increased competition, has influenced the adoption of EAI in OILCORP SUB_B;
- The existing IT infrastructure also influence the adoption of EAI solutions since its limitations prevent OILCORP SUB_B to achieve closer collaboration with its trading partners; and,
- The adoption of EAI solution requires changes at technical and organisational level.

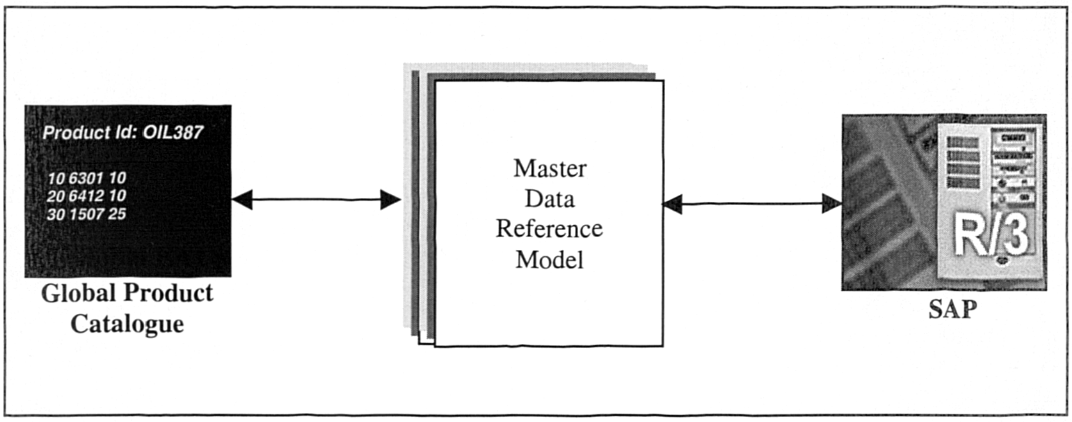

Figure 5.12: Pilot EAI Project Three

5.3.8 Global EAI project

The advantages (e.g. increased productivity, reduced lost sales) of the pilot systems One and Two were so significant that led the managing board to take the decision for a strategic adoption (Themistocleous and Irani, 2002)of application integration, before the end of EAI pilot projects Three. When the interviewees were asked to mention the factors that influenced this decision they reported the:

- benefits that derived from the pilot projects;
- cost of EAI solutions;
- level of technological problems that EAI addresses; and,
- impact on organisation, employees and trading partners.

It appears that the technical problems (e.g. ERPs incompatibility) that EAI overcomes are related with the existing IT infrastructure and its limitations. This is attributed to the organisation adopting EAI to address the problems that derive from the existing infrastructure (see sections 5.3.1-5.3.3). Therefore, it can be said that EAI benefits, cost, barriers and IT infrastructure were considered by OILCORP as factors that influences EAI adoption.

The global EAI project started in May 2001 and is estimated to finish in 2009. The project is divided in to two big phases: (a) the development of mega-datacentres and, (b) the integration of IT infrastructure. During the first phase three mega-datacentres will be developed in Europe (The Netherlands), America (USA) and Pacific-Asia (Malaysia). Each mega-datacentre will be based on a single SAP R/3 installation. All ERP systems from all subsidiaries will be phased out, with a transfer of their functionality to the regional ERP solution. Thus, only three SAP systems will be customised automated and integrated, based on the functionality and the processes of the existing 90 systems that are currently in use. In doing so, OILCORP will automatically phase out all 90 ERP systems after the development of these datacentres. This will dramatically reduced the number and redundancy of ERP applications used in the organisation. However, in achieving this solution, OILCORP needs to redesign and reengineer all its business processes.

OILCORP has signed a five-year €115 millions (£70 millions) single-source deal with IBM to provide hardware infrastructure. The first applications were migrated to the European mega-datacentre in October 2001, with the transfer of the other installations being completed in the next years. The overall cost for the development of the datacentres is estimated to be up to €575 millions (£350 millions) (€115 millions for hardware, €200 millions for software and €260 millions for development).

The second phase of the project will begin after the development of the mega-datacentres. It is estimated that once the European datacentre has been implemented, the integration phase will commence. As illustrated in Figure 5.13, the global EAI solution is geographically divided in three big integrated subsystems. Each of these regional subsystems integrates the applications and processes from all subsidiaries that are based in its region. For instance, all European subsidiaries of OILCORP should integrate their applications using the European hub and spoke infrastructure. In addition to the three regional hub and spoke integration infrastructures, a global one will be developed to unify the regional integrated infrastructures.

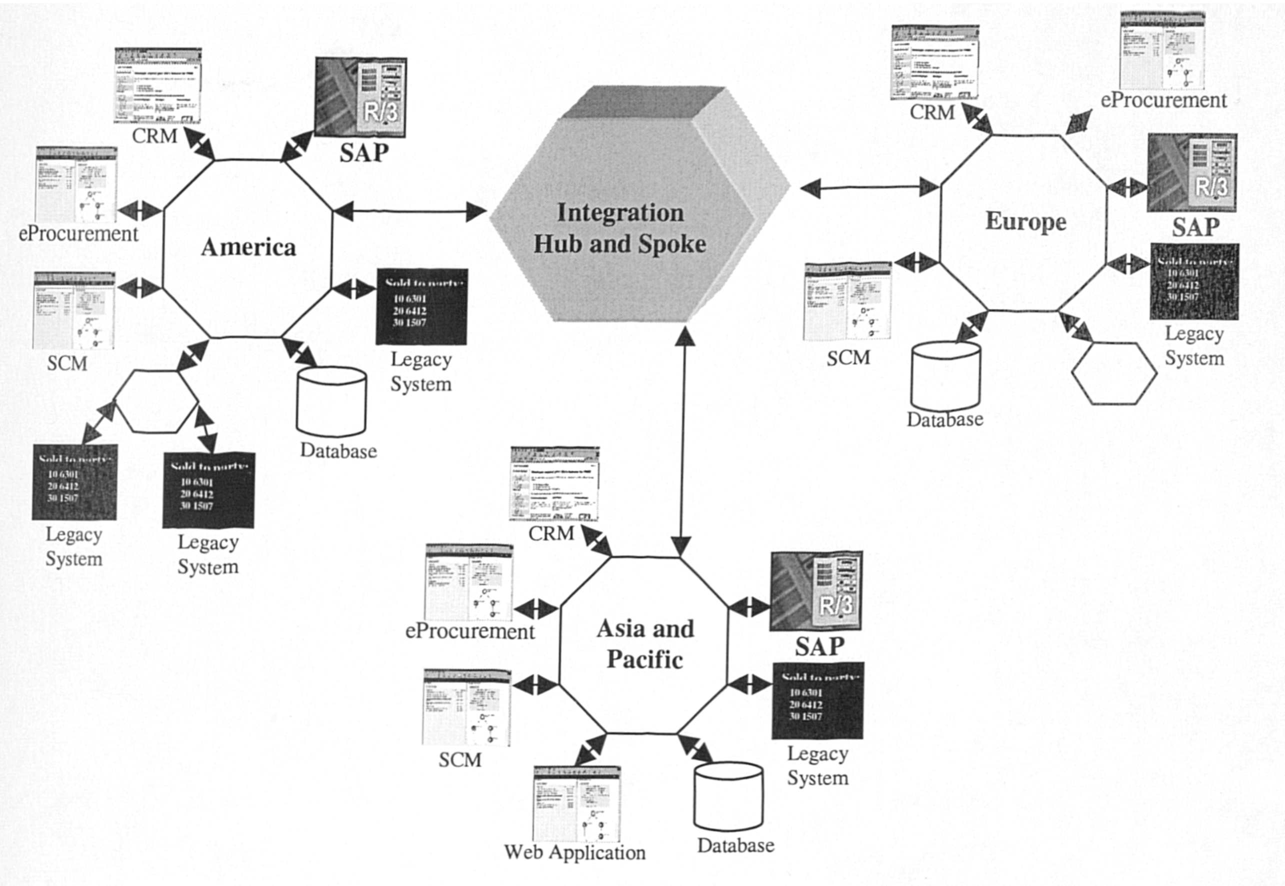

Figure 5.13: The Global Application Integration Solution

Based on the model of EAI pilot project One, each regional subsystem will integrate custom applications (legacy, databases, data warehouses etc) and e-business solutions with SAP modules. Regional subsystems may have a number of hubs to integrate all applications. It is estimated that 90% of custom systems will also be phased out with only 150 legacy systems remaining operable around the globe. The global EAI solution will need much business process reengineering. Based on the experience from pilot projects, OILCORP estimates that 60% of overall time will be needed for the design of the system. It is also estimated that 300 employees will work on the global EAI project, and will absorb a big portion of the total budget, which is estimated up to €64.6 millions [£40 millions] (only for integration). However, OILCORP estimates to have a €46 millions [£28 millions] savings from data consolidation (e.g. less maintenance effort, data quality, less management and technical effort). Figure 5.14 illustrates in more details the e-business integrated architecture.

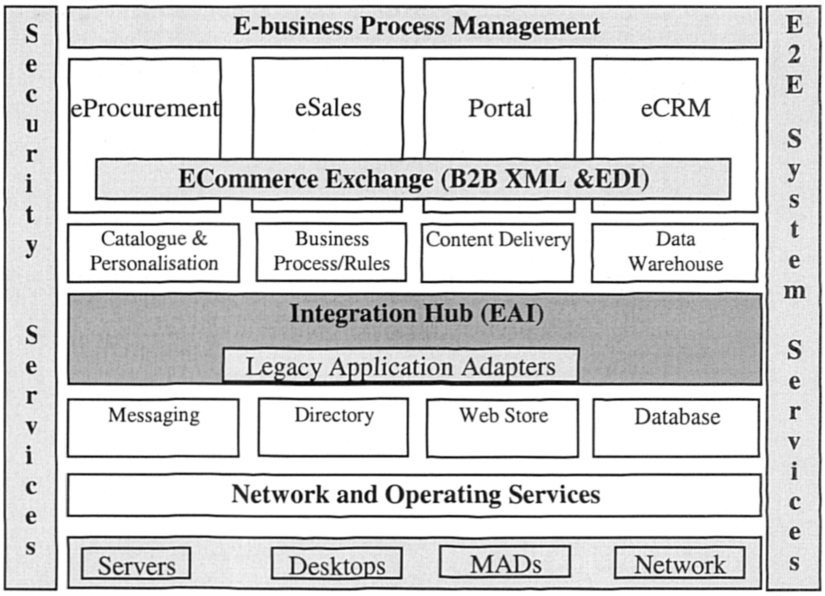

Figure 5.14: OILCORP - e-business Integration at Global EAI Project

5.3.9 Benefits

The benefits reported in this section reflect benefits that are derived from the EAI pilot projects One and Two. Also, this section summarises interviewees' estimations regarding the benefits that global EAI will provide the organisation. The benefits of application integration are classified using the model proposed by Shang and Seddon (2000) with the first two columns of Table 5.22 presenting the category of EAI benefits and the EAI benefits respectively.

Interviewees share similar perceptions and reported that the integrated infrastructure resulted from the implementation of the EAI pilot project One and Two increased organisation's performance and systems efficiency and functionality. It clearly automated business processes and integrated applications. The integration provided more understanding and control of business processes, with activities having been improved through reengineering. The integrated infrastructure reduces the redundancy of systems, objects and data and eliminates manual integration tasks. The pilot system provides more reliable data and is more flexible as integration was achieved with minimum changes to the systems code. As a result, the systems are more manageable and maintainable.

Category	Application Integration Benefits	I.C	Int.	P.M.
Operational	Improves planning in supply chain management	●	●	●
	Increases productivity	-	●	●
	Reduces cost	●	●	●
	Quicker response to change	●	●	●
	Reduces lost sales	-	-	●
Managerial	Provides more understanding and control of processes	-	●	●
	Results in more organised business processes	●	●	●
	Increases performance	-	●	●
	Improves data quality	◒	●	◒
	Improves management and supports decision making	●	◒	●
Strategic	Increases collaboration among partners	●	●	●
	Achieves customer satisfaction	●	◒	●
	Achieves return on investment	●	-	●
Technical	Offers interfaces-standardisation	-	●	●
	Results in reusable systems, components and data	◒	●	●
	Reduces redundancy of applications, data and tasks	◒	●	●
	Faster and cheaper implementation than bespoke solutions	●	●	●
	Increases flexibility	●	●	●
	Provides flexible, maintainable and manageable solutions	●	●	●
	Provides portability	-	◒	◒
	Reduces development risks	◒	●	◒
	Achieves non-invasive solutions	●	●	●
	Achieves process integration	●	●	●
	Increases data analysis	●	●	●
	Supports efficient data sharing	●	●	●
	Results in reliable data	-	●	●
	Process and systems scalability	◒	●	◒
Organisational	Allow organisations to do business more effectively	●	●	●

Table 5.22: Classification of OILCORP's EAI Benefits

5.3.10 Barriers

During the implementation phase, a number of problems have arisen regarding application integration. OILCORP has to consider and deal with political issues and the demands of its subsidiaries and, take important decisions. OILCORP SUB_C is an IT company that focuses on supporting the integration efforts of OILCORP and its subsidiaries. Nonetheless, the development of an integrated IT infrastructure will dramatically reduce OILCORP's SUB_C income since its market (OILCORP group) will be eliminated. The reason for this is that, after the adoption of an EAI infrastructure, the organisation will be less depended on technical support provided by OILCORP SUB_C. This causes much resistance by OILCORP's SUB_C, which attempted to persuade its OILCORP that, EAI is not mature and thus, a bespoke solution based on point-to-point interconnections will be the best for OILCORP. In case OILCORP had made a decision for adopting a point-to-point solution, OILCORP SUB_C would have had a tremendous benefit from the implementation and the maintenance of this solution. This is attributed to that OILCORP SUB_C would implement such a system in this case.

In the past, there was a lot of diversity around ERP implementations as OILCORP subsidiaries made a strong case for localisation. This individualism is a problem, as operating units are now sceptical about integration, as they fear that they may loose autonomy. It therefore appear that operating units resist to EAI adoption due to political reasons (e.g. loose autonomy). Although integrated architecture will help both OILCORP and its companies, some of its subsidiaries prefer to have their own solutions and not to share data and processes. In other cases, there is security scepticism, as subsidiaries hesitate to allow customers and partners to access their applications through e-business networks (internet, extranet).

There is another problem related to employees' and partners' resistance to change. Partners do not want to change their way of doing business, and integrate their IT infrastructure. However, some partners will be forced to change, as the majority of their transactions are financially dependent on (or related to) OILCORP group. Interviewees estimated that this category of partners would soon adopt application integration solutions too. This presents pressure from OILCORP to its trading partners to adopt EAI and it appears to form a factor that influences the adoption of EAI by trading partners.

Furthermore, OILCORP must face its employees' resistance to change. A lot of employees fear that their tasks and duties will be reduced through integrated systems and therefore, OILCORP will gradually employs fewer people. Other employees fear that they will lose their power, as their job will be less important to the group. Furthermore, employees with old technical skills fear that they will not be able to operate in an integrated environment, as they are short of hard skills.

The managing board believed that education will help employees understanding the reasons for adopting integrated solutions, and thus, reduce resistance to change. Training will also help employees adapt to using a system, as they will advance their knowledge and learn how to handle and operate an integrated solution. However, the training and the transition to the integrated system will cost a great amount of money. Table 5.23 classifies barriers according to Shang and Seddon's (2000) model.

Category	Application Integration Barriers	I.C	Int.	P.M.
Operational	Extra cost for redesign and change business structure, processes	●	●	●
Managerial	High complexity in understanding the processes and systems in order to redesign and integrate them	●	●	●
	Complexity of business processes	●	●	●
	Earlier approaches on EAI had proved problematic	-	◒	◒
Strategic	Politics issues	●	●	●
	Political impact (e.g. who controls the processes)	●	●	●
Technical	No single EAI product solves all integration problems	●	●	●
	Lack of employees with EAI skills	●	●	●
	EAI has a high cost	●	◒	●
Organisational	Resistance to change	●	●	●
	No time for training employees on integration technologies	◒	●	●
	Cultural issues	●	●	●

Table 5.23: Classification of OILCORP's EAI Barriers

5.3.11 Costs

The total cost for application integration at OILCORP was initially estimated to be globally more than €246 millions [£150 millions]. However, this cost has since been modified, as a result of OILCORP taking the decision for a global EAI solution that is based on three mega-datacentres. As reported in section 5.3.8 the cost for the development of the three datacentres was up to €575 millions [£350 millions]. The cost for integrating the organisation is estimated up to €65 millions [£40 millions]. This cost does not include the mega-datacentres development costs since the latter are not considered as part of EAI project. However, the cost of a solution based on EAI is much cheaper comparing to the cost of developing point-to-point interconnections. The latter was estimated to €246 millions [£150 millions]. It appears that one of the reasons that led OILCORP to adopt an EAI solution and not a point-to-point was its cost with the project manager reporting:

> *"There is no doubt that cost influenced the decisions to adopt EAI. Comparing to point-to-point solutions EAI is cheaper. If you also take into consideration the benefits and the impact of EAI, its is much more effective than point-to-point interconnectivity... Cost also influenced the decision of purchasing EAI products. For example we chose Crossworlds [software] because it supports a good level of integration and is cheaper than other solutions."*

However, cost was not the only factor that influenced the adoption of EAI in OILCORP. Decision-making was based around other parameters such as benefits, barriers, and as reported in previous sections, on existing IT infrastructure and the existence of an evaluation framework.

The costs of integration at OILCORP (excluding costs for datacentres development) was summarised to include:

- The cost of software and hardware for the development of integrated architecture;
- The cost for maintaining the software (e.g. licenses);

- The adoption of an integrated infrastructure is strongly related to OILCORP group's business strategy. The organisation took the decision to integrate both internal and external applications;
- To be more customer driven and to increase its competitive advantages. As a result, OILCORP has an extra cost for adapting the organisation and its strategies into the new environment, which has resulted in radical changes to organisation, and business strategies;
- The cost for redesigning and changing business processes allowed them to become more customer driven;
- Development and consultancy costs. These costs related to the implementation of integration hub as well as with the incorporation of the existing systems. Existing systems have been redesigned and changed to be customer driven. In cases that systems could not be changed to meet this requirement they were phased out and new applications are developed to replace them; and,
- The high costs for changing the way people currently work and include organisational changes and training. Part of this cost is related to staff development, and allowing employees to advance their technical skills. In addition, other training costs are related to the adapting of staff in the system (system usage training).

Category	Application Integration Costs	I.C	Int.	P.M.
Direct Costs	Hardware costs	○	○	○
	Software costs	●	●	●
	Development costs	●	●	●
	Maintenance costs	◒	◒	◒
	Consultancy costs	●	●	●
Indirect Human Costs	Employees training	◒	-	◒
	Changing employees culture	●	-	●
	Management efforts	●	-	●
Indirect Organisational Costs	Business Process re-engineering	●	●	●
	Organisational restructuring	●	●	●
	Covert resistance	●	-	●
	Strategy redesign	●	-	●

Table 5.24: Classification of OILCORP's EAI Costs

As reported by all interviewees, hardware costs that are primary associated with the implementation of EAI and do not cost too much compared to software and development costs. In addition, it seems that in the case of OILCORP the organisation will spend large amounts of money on business process redesigning and organisation restructuring.

5.4 Conclusions

This chapter presents and analyses the enterprise application integration adoption practice in two multinational organisations. The justification for selecting only two case companies is reported in section 3.1 and it is based on the rational that these two case studies have provided enough data for this research. As a result, the author suggests that a third case study would provide marginal benefits.

Based on the empirical data reported in this chapter, the enquiry now is able to draw conclusions. However, before any conclusions can be presented, it is important to appreciate the positioning of such conclusions within the context of the empirical research methodology presented in the preceding chapter. Chapter 4 has distinguished the research issues within the dissertation, with the conclusions presented in this chapter now forming level 3 as presented in Table 4.4. As a result, the following represents those conclusions derived from the empirical research presented in this chapter.

Empirical data revealed from these case studies confirm that the factors proposed in the conceptual model in section 3.6 have affected the adoption of EAI in AUTOCORP and OILCORP. Additionally, other factors that influence EAI adoption have been reported such as internal pressures and a framework for the evaluation of EAI packages. Modifications to the proposed framework for the assessment of integration technologies have also been reported. All the factors that have been reported and analysed in this chapter are taking into consideration in Chapter 6 to revise the conceptual model proposed in section 3.6. The key conclusions elicited from the adoption and evaluation of EAI in OILCORP and AUTOCORP are summarised in the following paragraphs.

- The tremendous changes in the business arena require flexible organisations that can easily adapt to the needs of the business environment. Both OILCORP and AUTOCORP could not effectively address this goal (flexibility in adapting to the changing business environment) since their existing IT infrastructure was an obstacle. This is attributed to that their IT infrastructures consist of hundreds of incompatible information systems, which could not fully automate and integrate their business processes. This limitation also increases maintenance cost and effort and does not effectively support decision-making.

- Pressures from the business environment such as increased competition and trading partners' pressures appear to have affected the decisions of OILCORP and AUTOCORP for EAI adoption. Empirical data indicate that both organisations have moved to EAI adoption to address: (a) increased competition and, (b) the pressures from their partners for closer collaboration. There is therefore, evidence to suggest that trading partners and competition form a factor (external pressures) that influences EAI adoption decisions in both organisations.

- The cases presented in this chapter also suggest that companies considered the EAI benefits, barriers and costs before adopting EAI. To better identify these factors (e.g. benefits) and assess the impact of EAI, both organisations took the decision to run pilot EAI projects. The reasons for running EAI pilot projects are the following:

    - The adoption of a global EAI solution requires considerable amounts of money. Both organisations were hesitant to invest these amounts of money without assessing the risks and the impact that associated with this investment;
    - There was a lack of knowledge regarding the drawbacks associated with the introduction of an EAI solution. This lack made the organisations reluctant to take a decision for EAI adoption without understanding the impact of this decision; and,
    - The benefits deriving from an integrated IT infrastructure could not estimated due to lack of knowledge associated with EAI.

These pilot projects offered valuable information regarding the EAI and supported AUTOCORP and OILCORP in decision-making for EAI adoption. Thus, it appears that benefits, cost, and barriers can be considered as influential factors for EAI adoption.

- The case studies reported in this chapter indicate that IT infrastructure is a factor for EAI adoption. The reason for this is that the adoption and the design of an EAI solution will be based on the systems types and the existing IT infrastructure. The existing IT infrastructure also appears to influence the selection of appropriate EAI packages and integration technologies.

- Another factor associated with the adoption and evaluation of EAI is IT sophistication. Empirical evidences indicate that IT sophistication affects the decisions for EAI adoption. In both organisations, there was a low level of IT sophistication regarding integration area and EAI. Therefore, AUTOCORP and OILCORP collaborated with consultants to improve their IT sophistication and support EAI adoption.

- Empirical evidence elicited from both case studies show that the existence of evaluation frameworks supported decision-making regarding the evaluation and selection of appropriate EAI packages and integration technologies. Case data indicate that the use of an evaluation framework for the assessment of EAI packages has an important role during the adoption and evaluation of EAI. There is therefore evidence suggesting that both: (a) a framework for the evaluation of integration technologies and, (b) a framework for EAI packages assessment, consist factors that influence the decisions for EAI adoption and evaluation. Both of these frameworks resulted in the selection of appropriate technologies and packages for EAI adoption. Thus, they influenced the decision making process for EAI evaluation and adoption.

- Support is another factor that appears to affect the introduction of enterprise application integration in both organisations. Empirical evidences suggest that support factors deal with vendors support and vendors' global presence are taken into consideration when evaluating integration technologies and EAI packages. Consultants' support was important in both AUTOCORP and OILCORP since it supported these organisations to better understand the capabilities of EAI and resulted in improved IT sophistication. Therefore, it influenced the decision making process for EAI adoption and evaluation.

Chapter 6: Enterprise Application Integration Adoption Model

Summary

The preceding chapter explored the research issues identified in Chapter 3, which dealt with the factors that influence the adoption of EAI. In doing so, Chapter 5 presented and analysed case studies, which were conducted in two multinational organisations. The issues in practice and the empirical evidence that resulted from the analysis indicate the need for modifications to the conceptual model proposed in Figure 3.3. This chapter takes into consideration the empirical data to revise the conceptual model. In doing so, satisfying the aim of this dissertation by offering decision-makers and researchers a frame of references for the *adoption and evaluation of enterprise application integration*.

6.1 Introduction

There has been much discussion around the lack of suitable theoretical models for the adoption of enterprise application integration. Indeed, the literature presented in Chapter 2 has emphasised this point, and it echoed this through empirical evidence reported in Chapter 5. In doing so, this thesis has investigated management concerns, and proposed the identification of factors that influence the adoption of EAI. Therefore, contributing towards a better understanding of the process associated with the adoption and evaluation of enterprise application integration.

Chapter 5 has offered much empirical data that has been used to assess the conceptual model presented in section 3.6 (see Figure 3.3), for the adoption of EAI. The aim of this chapter is to take into consideration the empirical data that derived from the previous chapter and offer revisions to the conceptual model for EAI adoption. The following section summarises the key issues that elicited from the analysis of the case studies. In doing so, exemplifying their need to be considered in any revisions to the conceptual morel. Thereafter, section 6.3 revises the conceptual model for EAI adoption and the proposed framework for evaluating integration technologies. Modifications are made to the conceptual model by adding two new factors that are derived from empirical research. These factors focus on: (a) the internal pressures and, (b) a framework that supports the evaluation of EAI packages. Modifications are also made on the framework for the assessment of integration technologies proposed in section 3.4. These modifications are derived from empirical evidences and deal with the addition of three new evaluation criteria namely: (a) connectivity integration layer; (b) security and, (c) manageability.

Hence, this chapter results in the proposition of a novel conceptual model for the adoption and evaluation of EAI technology. Such a model can be used by organisations as a tool for decision-making when adopting EAI.

6.2 Lessons Learned from Case Studies

A synopsis of the main findings elicited from Chapter 5 is given in this section. In doing so, allowing others to relate their experiences to those reported in Chapter 5. It is not the intention of this section (or this dissertation) to offer prescriptive guidelines to EAI adoption and evaluation but rather, describe case study perspectives that allow others to relate their experiences to those reported. Hence, this dissertation offers a broader understanding of the phenomenon of enterprise application integration adoption.

A number of parameters have been extrapolated from the empirical data and identified as factors that were taken into consideration during the adoption of EAI by the two organisations reported. The key issues that derived from empirical data are summarised below:

- The cases of OILCORP and AUTOCORP presented in Chapter 5 revealed that these organisations turned to EAI for many reasons including, technical, managerial, financial and strategic motivations. Clearly, these factors present internal pressures to the adoption of enterprise application integration. *However, such factors were not included in the conceptual model proposed in Chapter 3 and thus, additions are required to the aforementioned model to reflect these empirical findings.*

- Empirical data indicated that competitors' pressure was another driver for EAI introduction at OILCORP. AUTOCORP has also moved to EAI adoption to address both competitors and trading partners' pressures. These two factors represent external pressures for EAI adoption.

- In both cases, the cost of EAI adoption, the high risk and the lack of reflective learning EAI cases[1] influenced the decisions of OILCORP and AUTOCORP to adopt EAI. These issues deal with cost and barriers and affect the introduction of EAI in organisations. A lack of sufficient knowledge of enterprise application integration is also related to IT sophistication, since organisations could not

[1] The lack of reflective EAI cases increases the risk since there is not sufficient knowledge regarding: (a) the problems associated with EAI implementation and, (b) the impact of EAI adoption.

understand the capabilities of EAI. Cost, barriers and IT sophistication are factors that are included in the conceptual model proposed in Chapter 3. However, the cases studied provided enough data to classify the costs into (a) direct; (b) indirect human and, (c) indirect organisational costs. In doing so, continuing the application of the cost model proposed by Hochstrasser (1992) when used for the purpose of adoption of EAI. Such consideration of cost factors increases cost analysis, allow better understanding and contribute to better decision making. Hence, *direct, indirect human and indirect organisational cost factors are added in the conceptual model.*

- Although the managing board of both organisations rejected the initial proposals for a global EAI adoption they supported the implementation of pilot projects. The reasons for this decision were that: (a) insufficient IT sophistication was an obstacle to the adoption of EAI and, (b) each organisation needed to evaluate the impact (benefits, barriers, costs, integrated IT infrastructure etc) of EAI before proceeding to a global adoption. The former indicates that IT sophistication was a factor that affected the introduction of EAI in these organisations. The latter shows that both OILCORP and AUTOCORP needed to assess the level of integration that is achieved using EAI, as well as to evaluate its benefits barriers and costs. All these parameters represent factors that influence EAI adoption and were included in the conceptual model proposed in Chapter 3.

- OILCORP and AUTOCORP took the decision to evaluate EAI packages and integration technologies before the implementation of the pilot EAI projects. Both organisations have developed an evaluation framework for the assessment of integration technologies. It appears that these organisations invested money and time to develop their frameworks, understand and evaluate EAI technologies. Moreover, they believe that such a framework is an important decision-making tool that influenced their decisions to adopt EAI.

- Empirical evidence indicates extensions to the framework for the evaluation of integration technologies proposed in Chapter 3. *Both organisations suggested additional criteria for the proposed framework* that dealt with: (a) security; (b) manageability; (c) connectivity layer (d) vendor support and, (e) vendor global presence. The last two criteria are related to support factors, which are also included in the proposed revised conceptual model. Moreover, AUTOCORP reported that the cost of technology is an extra evaluation criterion. Nonetheless, costs are included as a factor at the conceptual model and thus, no further additions are required regarding this issue. Therefore, only *the criteria that refers to: (a) security; (b) manageability; and, (c) connectivity layer,* need to be added to the proposed evaluation framework.

- In addition to the framework for evaluating integration technologies, both organisations developed and used individual frameworks for the assessment of EAI packages. OILCORP did not report the evaluation criteria for the selection of EAI (packages) due to confidentiality reasons. Findings reported in the case of AUTOCORP are along similar lines to the literature (Ring and Ward-Dutton, 1999), and indicate that the following criteria were considered when evaluating EAI packages: (a) integrated Vs toolkit EAI; (b) tight Vs loose integration; (c) individual (or custom) Vs standard (or packaged) and, (d) intra Vs Inter-organisational integration. *These findings suggest additions to the conceptual model. In doing so, a new evaluation framework that supports the assessment of EAI packages should be incorporated to the proposed revised model.*

- After the implementation of the pilot projects OILCORP and AUTOCORP took the decision to develop a global integrated IT infrastructure. The overall benefits that derived from the implementation of the pilot projects at both organisations have led them to a global EAI adoption. As reported by interviewees, this decision was based on the assessment of: (a) EAI benefits; (b) EAI barriers (and impact); (c) costs; (d) the nature of the technological solution that EAI supports; (e) the IT sophistication to understand and solve their integration problems through EAI and, (g) support factors. These evidences establish the conceptual model proposed in Chapter 3.

Table 6.1 summarises the additions to the conceptual model as derived from the aforementioned case studies, which in turn suggests changes in the conceptual model.

Factors	OILCORP	AUTOCORP
Costs		
Direct costs	✓	✓
Indirect human costs	✓	✓
Indirect organisational costs	✓	✓
Internal Pressures		
Technical	✓	✓
Managerial	✓	✓
Operational - Financial	✓	✓
Strategic	✓	✓
Organisational	✓	✓
Framework for Evaluating Integration Technologies		
Security	✓	✓
Manageability	✓	✓
Connectivity integration layer	✓	✓
Framework for Evaluating EAI Packages		
Integrated application	-	✓
Toolkit application	-	✓
Loosely coupling	-	✓
Tightly coupling	-	✓
Custom systems integration	-	✓
Packaged systems integration	-	✓
Intra-organisational integration	-	✓
Inter-organisational integration	-	✓

Table 6.1: Additional Factors Influencing EAI Adoption Derived from Empirical Evidences

6.3 The Revised Conceptual Model for EAI Adoption

The process of developing Figure 6.1, which is EAI specific, has only been made possible after having carried out the empirical research reported in Chapter 5. As a result, following the investigation of research issues identified in Chapter 3, a revised conceptual model is now presented Figure 6.1.

As illustrated in Figure 6.1 the adoption of enterprise application integration is influenced by ten factors namely: (a) costs; (b) barriers; (c) benefits; (d) internal pressures; (e) external pressures; (f) IT infrastructure; (g) IT sophistication; (h) support; (i) the existence of a framework for the evaluation of integration technologies and, (j) a framework for the assessment of EAI packages. These factors are now analysed in the following paragraphs.

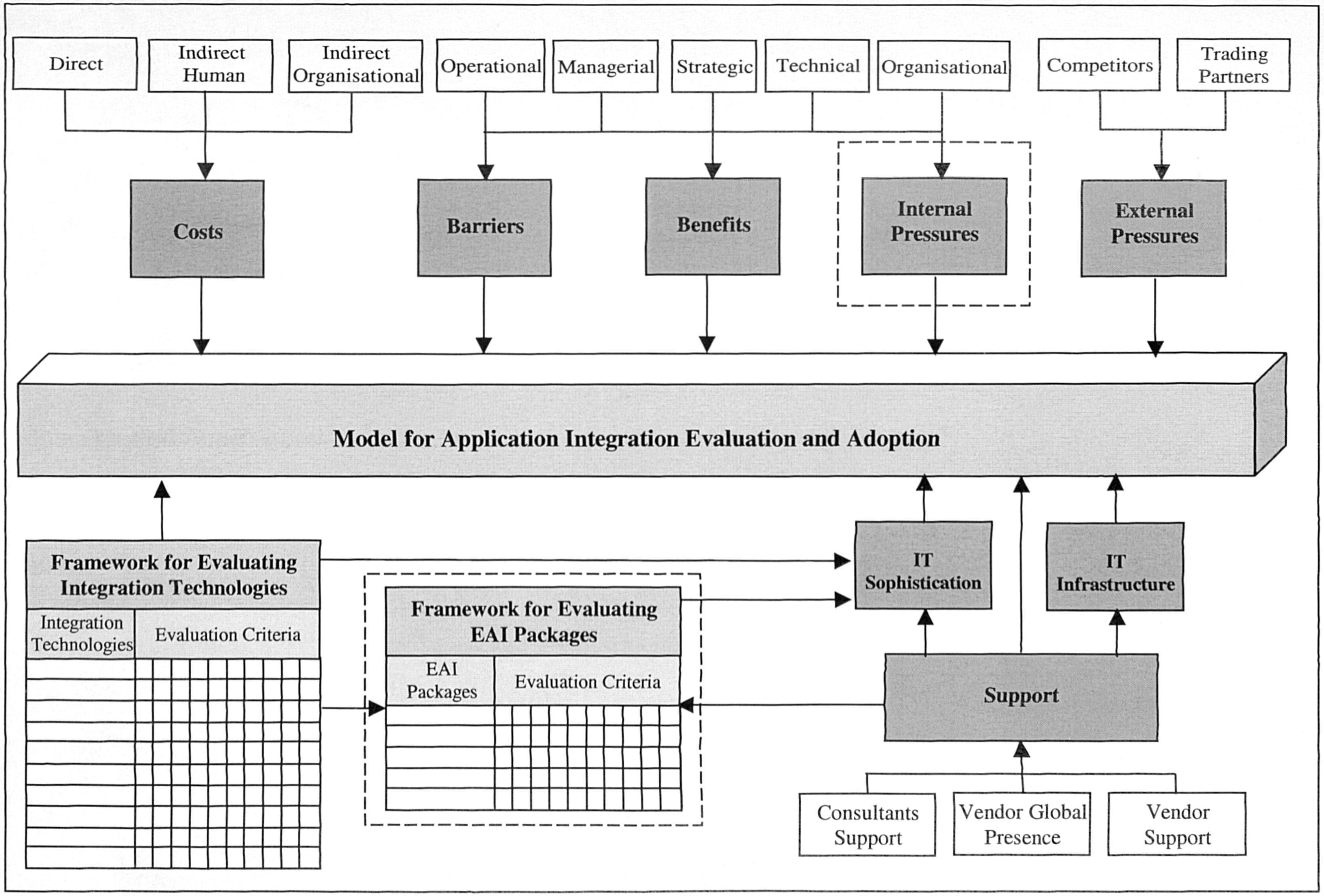

Figure 6.1: The Revised Conceptual Model for the Adoption and Evaluation of EAI

In Figure 6.1, the new factors that were derived from empirical evidences are included into a dashed box (e.g. internal pressures). The highlighted text boxes (e.g. costs, framework for evaluating EAI packages) consist factors that influence the adoption and evaluation of EAI. Text boxes that have no shadow (or colour) consist sub-factors (e.g. direct, indirect human and indirect organisational costs are sub-factors of cost factor).

6.3.1 Costs

The cases of the two multinational organisations presented in Chapter 5 have shown that EAI adoption required substantial investment of organisational funds. The considerable cost of investment had led both organisations to justify and evaluate the implications of the introduction of EAI. In doing so, indicated that cost is a significant parameter that influences the adoption of enterprise application integration. This finding is in accordance with literature (Irani *et al.*, 1997; 1998) which supports that organisations justify their costs and benefits before the adoption of a new investment. Cost as a factor that affects the adoption of integration technologies such EDI is also reported in studies such as Chwelos *et al.*(2001). Also, costs influence the adoption of other technologies that were seen as integrated packages or suites (e.g. ERP systems) (Bernroider, 1999).

As mentioned in the previous section, costs are classified into: (a) direct; (b) indirect human and, (c) indirect organisational cost factors. According to Irani (1998), direct costs are those factors that can be attributed to the implementation and operation of IT/IS. In this case, direct costs refer to those costs that are related with: (a) the appropriate hardware for the development of an integrated IT infrastructure; (b) EAI software and other software categories (e.g. middleware, communication software) associated with integration; (c) development costs; (d) maintenance costs and, (e) consultancy costs. Indirect costs are financially tangible/intangible and non-financial in nature and can be divided into indirect human costs and indirect organisational costs. Indirect human costs include: (a) management efforts; (b) changing employees culture and, (c) employees training. Indirect organisational costs focus on costs such as (a) strategy redesign; (b) business process reengineering and, (c) organisational restructuring.

6.3.2 Barriers

Davenport (1998), Dong (2000) and Markus and Tanis (1999) support the view that the introduction of a new technology and especially ERP systems has often a negative impact (barriers) on organisations. Organisations need to estimate these barriers before taking their decisions to adopt a new technology. Davenport (1998) reports cases where the introduction of ERP systems has led organisations to bankruptcy as, they did not consider the impact of this technology on the organisation. An explanation to this failure is that ERP systems: (a) require high investments in IT infrastructure; (b) cause changes to the relationships between organisation and its trading partners; (c) result in employee resistance to change; (d) require skilled employees, (e) require changes in business strategy and, (f) changes in organisational structure.

Enterprise application integration clearly presents barriers with organisations needing to consider these barriers before proceeding to EAI adoption. Empirical evidences identified in Chapter 5 supports this view, with organisations hesitating to adopt EAI before estimating its impact (benefits, barriers, costs etc.). The cases of OILCORP and AUTOCORP have shown that EAI is related to a variety of barriers with the author classifying them into: (a) operational; (b) managerial; (c) strategic; (d) technical and, (e) organisational factors. Such a classification allows managers and researchers to better analyse and understand the barriers of EAI and results in increased decision-making.

6.3.3. Benefits

Organisations turn to enterprise application integration to improve their efficiency, productivity and integrate their business processes and systems. Although, EAI technology offers numerous advantages to organisations, such companies often prefer to assess these benefits before proceeding to EAI adoption. The cases of OILCORP and AUTOCORP have shown that these organisations justify the adoption of EAI by taking into consideration various parameters such as benefits, barriers, costs and qualities of technical solution provided by EAI. Thus, it appears that benefits consists factor that influences the adoption of enterprise application integration in organisations. This finding is in accordance with literature (Chwelos *et al.*, 2001; Markus and Tanis, 1999; Oliver and Romm, 2000) that considers benefits as a factor that affects the adoption of other areas like EDI and ERP. Both EDI and ERP systems were proposed as technologies that address integration problems, with the author drawing parallel between EDI-ERPs and EAI. Benefits were classified based on the model proposed by Shang and Seddon (2000) into: (a) operational; (b) managerial; (c) strategic; (d) technical and, (e) organisational factors. Tables 5.11 and 5.22 summarise EAI benefits based on the model proposed by Shang and Seddon (2000).

6.3.4. Internal Pressures

Empirical data have shown that internal pressures comprise as factor that affects the adoption of EAI. This factor refers to the various drivers that initiate the introduction of enterprise application integration. Internal pressures consist of the same sub-factors as benefits and barriers, and include among others the following:

- the insufficient nature of existing IT infrastructure (technical);
- the high cost of running and maintaining such an IT infrastructure (operational-financial);
- the limitations of existing systems (or the complexity of business processes) to support efficient decision-making (managerial);
- the obstacles that emerge from a non-integrated structure and infrastructure and which hold organisation to gain competitive advantages; and
- the organisational complexity and problems that a non-integrated infrastructure imposes (organisational).

6.3.5. External Pressures

This factor refers to the pressures that are imposed to the organisation by its environment. The tremendous changes in the business arena require flexible organisations that can easily adapt to change. Thus, organisations should be capable to respond to competitors' movements, as well as to their trading partners actions and requests. Both case studies presented in Chapter 5 indicated that external pressures had influenced the adoption of EAI by OILCORP and AUTOCORP. This is along the same lines with theoretical models such as that presented by Iacovou *et al.*(1995), which reports external pressures as factor that affects the adoption of a technology. External pressures consist of competitors' pressures and trading partners' pressures.

6.3.6. IT Sophistication

Chwelos *et al.*, (2001) report that IT sophistication influences the adoption of integrated technologies such as EDI. Information technology sophistication is related to the level of understanding and addressing technical problems within the organisation. More specifically, IT sophistication refers to: (a) the level of understanding technical problems (e.g. what the integration limitations of existing IT infrastructure are); (b) the level of knowledge regarding the capabilities of integration technologies and, (c) the level of understanding enterprise application integration (e.g. how EAI can be used to achieve integration). In addition, IT sophistication deals with the people who have the technical skills to support an EAI project.

IT departments require a high level of IT sophistication regarding integration, since such a level allows an organisation to easily apprehend its integration problems and identify EAI solutions. Thus, a sufficient level of IT sophistication for EAI and integration technologies influences the adoption of enterprise application integration. The reason for this is that IT departments have a better understanding of both the organisation's integration problems, and available solutions for overcoming these problems.

The case studies analysed in Chapter 5 have shown that level of IT sophistication at OILCORP and AUTOCORP has affected the adoption of EAI. As reported in these cases, there was a lack of skilled employees at both organisations to understand integration problems or technologies. In support of this, both organisations collaborate with internal (individuals) or external consultants (companies) to improve IT sophistication. This indicates that support factors such as consultants influence the level of IT sophistication. Furthermore, as mentioned in sections 5.2.6.1 and 5.3.6.1 the existence of frameworks that support the evaluation of integration technologies and EAI packages has also improved the level of IT sophistication. This has caused a modification to the conceptual model proposed in chapter 3 and deals with that, the framework for EAI packages evaluation influences the IT sophistication.

6.3.7. IT Infrastructure

The IT infrastructure of an organisation consists a factor that influences the adoption of enterprise application integration. As reported in the case studies, the non-integrated nature of IT infrastructures at OILCORP and AUTOCORP has caused various problems to these companies (e.g. high operational and maintenance cost, low customer satisfaction etc.). The limitations of existing IT infrastructures have influenced the decisions to adopt EAI. However, the applications in use and the systems types of these IT infrastructures imposed the adoption of specific integration technologies and EAI packages. Hence, IT infrastructures influence the decision to adopt appropriate EAI solutions and integration software.

6.3.8. Framework for Evaluating Integration Technologies

Earlier sections of this dissertation (see sections 2.6, 2.8, 3.5, 5.2.6 and 5.3.6) have reported, analysed and investigated the issues that derive from the confusion that surrounds EAI marketplace and integration technologies. In doing so, section 2.8 has identified the need for the development of a framework for the evaluation of integration technologies. Section 3.5 proposed such a framework (see Table 3.4), which consists in part of the conceptual model (see Figure 3.3). The proposed framework was examined during the case studies with interviewees emphasising that:

- The proposed framework can be used as a decision-making tool and supports the adoption of integration technologies. In support of this, AUTOCORP expressed its intention to adopt the proposed framework, which indicates the importance of such a framework.

- Although the proposed framework supports decision-making regarding the evaluation and adoption of integration technologies, it requires minor modifications.

As reported in section 6.2, interviewees suggested six new criteria while the author uses only: (a) connectivity layer; (b) security and, (c) manageability, as additional criteria to the proposed framework. The other three criteria deal with support and costs factors that are included in the conceptual model.

Much of the discussion reported in Chapter 5 regarding the proposed framework, indicates that the evaluation framework should be restructured in a way that it allows decision-makers to classify evaluation results based on the integration layers. In support of this, the author divides the proposed framework in seven sub-frameworks with each of these presenting the evaluation results based on a system types (there exist seven system types [e.g. custom-packaged applications integration]). In addition, each sub-framework classifies integration technologies based on the four integration layers namely: (a) connectivity; (b) transportation; (c) translation and, (d) process automation.

Such a presentation of the proposed framework allows decision-makers to compare (horizontally) the technologies that can be used at each integration layer. Vertically, decision-makers can compare the qualities of each technology (e.g. whether it supports data, objects and process integration etc.). Moreover, decision-makers can more easily compare the diversity of technologies that supports the integration of each system type. The reason for this is that the evaluation of integration technologies for each system type is presented in a different sub-framework. Each of these sub-frameworks consists only of those technologies that support the integration of specific system type and therefore decreases complexity.

Table 6.2 shows the revised framework for the evaluation of integration technologies. This framework includes the additional criteria proposed during the case studies. Additional criteria are positioned at the right end of Table 6.2 and are highlighted. Tables 6.3, 6.4, 6.5, 6.6, 6.7, 6.8 and 6.9 present the revised sub-frameworks for the evaluation of integration technologies. Each of these sub-frameworks consists of: (a) the three categories of evaluation criteria presented in Table 3.3 (applications elements, integration layers and integration requirements) and, (b) only one system type of the fourth category of evaluation criteria (system types). Thus, these sub-frameworks present evaluation results based on: (a) custom-to-custom; (b) custom-to-packaged; (c) custom-to-e-business; (d) packaged-to-packaged; (e) packaged-to-e-business; (f) e-business-to-e-business and, (g) custom-to-packaged-to-e-business respectively. Each sub-framework incorporates the additional evaluation criteria that are derived from empirical data.

Although Tables 6.3 to 6.9 look alike, they are not. Each of these tables evaluates integration technologies based on a specific system type (e.g. custom-to-custom application integration). Each table classifies the integration technologies based on the integration layers that these technologies support. The author uses shading to separate the integration layers to better present the evaluation results. Integration technologies are evaluated using the criteria (e.g. data, objects, process) that are positioned at the top of each sub-framework.

The novelty of the proposed framework presented in Table 6.2 is that it is based on a comprehensive set of evaluation criteria. It contributes towards a better understanding of the capabilities of each technology, and allows decision-makers to clarify the confusion surrounding integration technologies. Such a framework can be used as a frame of references to highlight possible combinations of integration solutions that can address the integration of information systems. Also, the proposed framework improves IT sophistication since it contributes to understanding the capabilities of integration technologies.

	Evaluation Criteria																									Additional Criteria		
Integration Technologies	Maintainability	Flexibility	Scalability	Portability	Reusability	Maturity	Complexity	Non-invasive	Performance	Real Time	Mainframe Compatible	Non-Mainframe Compatible	Data	Objects	Process	Transportation	Translation	Process Automation	Custom to Custom	Custom to Packaged	Custom to e-business	Packaged to Packaged	Packaged to e-business	E-business to e-business	Custom to Packaged to e-business	Connectivity	Security	Manageability
ODBC																												
JDBC																												
RPC																												
MOM																												
Message Broker																												
XML																												
TPM																												
Application Serves																												
CORBA																												
DCOM / COM																												
EJB																												
Screen wrapper																												
APIs																												
Adapters																												

Table 6.2: The Revised Framework for the Evaluation of Integration Technologies

		Custom-to-custom Applications Integration																
		Data	Objects	Process	Maintainability	Flexibility	Scalability	Portability	Reusability	Maturity	Complexity	Non-invasive	Performance	Real Time	Mainframe Compatible	Non-Mainframe Compatible	Security	Manageability
Integration Layer	Technology																	
Process Automation	Message Broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
Translation	Message broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
	Adapters	●	●	×	●	✓	✓	✓	✓	○	○	✓	–	–	–	●		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
Transportation	MOM	✓	○	×	○	◐	○	×	×	◐	●	×	◐	×	●	●		
	TP Monitors	✓	×	×	–	○	●	●	–	◐	●	×	●	–	◐	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
Connectivity	ODBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	○	●		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
	Screen Wrappers	✓	✓	×	○	×	◐	–	✓	◐	●	✓	○	–	●	×		

Table 6.3: Evaluation Sub-Framework that Focuses on Custom-to-Custom Integration

		Custom-to-Packaged Applications Integration																
Integration Layer	Technology	Data	Objects	Process	Maintainability	Flexibility	Scalability	Portability	Reusability	Maturity	Complexity	Non-invasive	Performance	Real Time	Mainframe Compatible	Non-Mainframe Compatible	Security	Manageability
Process Automation	Message Broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
Translation	Message broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
	Adapters	●	●	×	●	✓	✓	✓	✓	○	○	✓	–	–	–	●		
	XML	✓	✓	×	◒	●	✓	●	–	◒	○	○	◒	✓	×	◒		
	CORBA	✓	●	–	◒	✓	◒	◒	●	○	–	×	◒	–	–	●		
Transportation	RPC	✓	×	×	○	×	×	–	✓	●	●	×	○	✓	●	●		
	MOM	✓	○	×	○	◒	○	×	×	◒	●	×	◒	×	●	●		
	XML	✓	✓	×	◒	●	✓	●	–	◒	○	○	◒	✓	×	◒		
	Application Servers	✓	✓	×	◒	◒	●	●	◒	○	◒	×	◒	–	–	◒		
	CORBA	✓	●	–	◒	✓	◒	◒	●	○	–	×	◒	–	–	●		
	COM/DCOM	✓	●	–	◒	✓	◒	○	●	○	○	×	◒	✓	×	◒		
Connectivity	ODBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	○	●		
	JDBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	×	◒		
	XML	✓	✓	×	◒	●	✓	●	–	◒	○	○	◒	✓	×	◒		
	CORBA	✓	●	–	◒	✓	◒	◒	●	○	–	×	◒	–	–	●		
	COM/DCOM	✓	●	–	◒	✓	◒	○	●	○	○	×	◒	✓	×	◒		
	Screen Wrappers	✓	✓	×	○	×	◒	–	✓	◒	●	✓	○	–	●	×		
	APIs	✓	✓	–	◒	✓	–	✓	✓	◒	●	✓	–	✓	×	●		

Table 6.4: Evaluation Sub-Framework that Focuses on Custom-to-Packaged Integration

Integration Layer	Technology	Custom-to-e-business Applications Integration																
		Data	Objects	Process	Maintainability	Flexibility	Scalability	Portability	Reusability	Maturity	Complexity	Non-invasive	Performance	Real Time	Mainframe Compatible	Non-Mainframe Compatible	Security	Manageability
Process Automation	Message Broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
Translation	Message broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
	Adapters	●	●	×	●	✓	✓	✓	✓	○	○	✓	–	–	–	●		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
Transportation	RPC	✓	×	×	○	×	×	–	✓	●	●	×	○	✓	●	●		
	MOM	✓	○	×	○	◐	○	×	×	◐	●	×	◐	×	●	●		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	Application Servers	✓	✓	×	◐	◐	●	●	◐	○	◐	×	◐	–	–	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
	COM/DCOM	✓	●	–	◐	✓	◐	○	●	○	○	×	◐	✓	×	◐		
	EJB	✓	●	–	◐	✓	◐	◐	✓	○	–	×	○	✓	×	◐		
Connectivity	ODBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	○	●		
	JDBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	×	◐		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
	COM/DCOM	✓	●	–	◐	✓	◐	○	●	○	○	×	◐	✓	×	◐		
	EJB	✓	●	–	◐	✓	◐	◐	✓	○	–	×	○	✓	×	◐		
	Screen Wrappers	✓	✓	×	○	×	◐	–	✓	◐	●	✓	○	–	●	×		

Table 6.5: Evaluation Sub-Framework that Focuses on Custom-to-e-business Integration

		Packaged-to-Packaged Applications Integration																
		Data	Objects	Process	Maintainability	Flexibility	Scalability	Portability	Reusability	Maturity	Complexity	Non-invasive	Performance	Real Time	Mainframe Compatible	Non-Mainframe Compatible	Security	Manageability
Integration Layer	**Technology**																	
Process Automation	Message Broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
Translation	Message broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
	Adapters	●	●	×	●	✓	✓	✓	✓	○	○	✓	–	–	–	●		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
Transportation	MOM	✓	○	×	○	◐	○	×	×	◐	●	×	◐	×	●	●		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	Application Servers	✓	✓	×	◐	◐	●	●	◐	○	◐	×	◐	–	–	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
	COM/DCOM	✓	●	–	◐	✓	◐	○	●	○	○	×	◐	✓	×	◐		
	EJB	✓	●	–	◐	✓	◐	◐	✓	○	–	×	○	✓	×	◐		
Connectivity	ODBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	○	●		
	JDBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	×	◐		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
	COM/DCOM	✓	●	–	◐	✓	◐	○	●	○	○	×	◐	✓	×	◐		
	EJB	✓	●	–	◐	✓	◐	◐	✓	○	–	×	○	✓	×	◐		
	APIs	✓	✓	–	◐	✓	–	✓	✓	◐	●	✓	–	✓	×	●		

Table 6.6: Evaluation Sub-Framework that Focuses on Packaged-to- Packaged Integration

		Packaged-to-e-business Applications Integration																
		Data	Objects	Process	Maintainability	Flexibility	Scalability	Portability	Reusability	Maturity	Complexity	Non-invasive	Performance	Real Time	Mainframe Compatible	Non-Mainframe Compatible	Security	Manageability
Integration Layer	**Technology**																	
Process Automation	Message Broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
Translation	Message broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
	Adapters	●	●	×	●	✓	✓	✓	✓	○	○	✓	–	–	–	●		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
Transportation	MOM	✓	○	×	○	◐	○	×	×	◐	●	×	◐	×	●	●		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	Application Servers	✓	✓	×	◐	◐	●	●	◐	○	◐	×	◐	–	–	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
	COM/DCOM	✓	●	–	◐	✓	◐	○	●	○	○	×	◐	✓	×	◐		
	EJB	✓	●	–	◐	✓	◐	◐	✓	○	–	×	○	✓	×	◐		
Connectivity	ODBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	○	●		
	JDBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	×	◐		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
	COM/DCOM	✓	●	–	◐	✓	◐	○	●	○	○	×	◐	✓	×	◐		
	EJB	✓	●	–	◐	✓	◐	◐	✓	○	–	×	○	✓	×	◐		
	APIs	✓	✓	–	◐	✓	–	✓	✓	◐	●	✓	–	✓	×	●		

Table 6.7: Evaluation Sub-Framework that Focuses on Packaged -to-e-business Integration

Integration Layer	Technology	Data	Objects	Process	Maintainability	Flexibility	Scalability	Portability	Reusability	Maturity	Complexity	Non-invasive	Performance	Real Time	Mainframe Compatible	Non-Mainframe Compatible	Security	Manageability
		e-business-to-e-business Applications Integration																
Process Automation	Message Broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
Translation	Message broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
	Adapters	●	●	×	●	✓	✓	✓	✓	○	○	✓	–	–	–	●		
	XML	✓	✓	×	◒	●	✓	●	–	◒	○	○	◒	✓	×	◒		
	CORBA	✓	●	–	◒	✓	◒	◒	●	○	–	×	◒	–	–	●		
Transportation	MOM	✓	○	×	○	◒	○	×	×	◒	●	×	◒	×	●	●		
	XML	✓	✓	×	◒	●	✓	●	–	◒	○	○	◒	✓	×	◒		
	Application Servers	✓	✓	×	◒	◒	●	●	◒	○	◒	×	◒	–	–	◒		
	CORBA	✓	●	–	◒	✓	◒	◒	●	○	–	×	◒	–	–	●		
	COM/DCOM	✓	●	–	◒	✓	◒	○	●	○	○	×	◒	✓	×	◒		
	EJB	✓	●	–	◒	✓	◒	◒	✓	○	–	×	○	✓	×	◒		
Connectivity	ODBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	○	●		
	JDBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	×	◒		
	XML	✓	✓	×	◒	●	✓	●	–	◒	○	○	◒	✓	×	◒		
	CORBA	✓	●	–	◒	✓	◒	◒	●	○	–	×	◒	–	–	●		
	COM/DCOM	✓	●	–	◒	✓	◒	○	●	○	○	×	◒	✓	×	◒		
	EJB	✓	●	–	◒	✓	◒	◒	✓	○	–	×	○	✓	×	◒		
	APIs	✓	✓	–	◒	✓	–	✓	✓	◒	●	✓	–	✓	×	●		

Table 6.8: Evaluation Sub-Framework that Focuses on e-business-to- e-business Integration

Integration Layer	Technology	Data	Objects	Process	Maintainability	Flexibility	Scalability	Portability	Reusability	Maturity	Complexity	Non-invasive	Performance	Real Time	Mainframe Compatible	Non-Mainframe Compatible	Security	Manageability
		Custom-to-packaged-to-e-business Applications Integration																
Process Automation	Message Broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
Translation	Message broker	●	○	●	●	●	●	●	●	○	○	✓	●	○	●	●		
	Adapters	●	●	×	●	✓	✓	✓	✓	○	○	✓	–	–	–	●		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
Transportation	RPC	✓	×	×	○	×	×	–	✓	●	●	×	○	✓	●	●		
	MOM	✓	○	×	○	◐	○	×	×	◐	●	×	◐	×	●	●		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	TP Monitors	✓	×	×	–	○	●	●	–	◐	●	×	●	–	◐	◐		
	Application Servers	✓	✓	×	◐	◐	●	●	◐	○	◐	×	◐	–	–	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
	COM/DCOM	✓	●	–	◐	✓	◐	○	●	○	○	×	◐	✓	×	◐		
	EJB	✓	●	–	◐	✓	◐	◐	✓	○	–	×	○	✓	×	◐		
Connectivity	ODBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	○	●		
	JDBC	●	×	×	–	–	–	✓	✓	●	○	–	●	–	×	◐		
	XML	✓	✓	×	◐	●	✓	●	–	◐	○	○	◐	✓	×	◐		
	CORBA	✓	●	–	◐	✓	◐	◐	●	○	–	×	◐	–	–	●		
	COM/DCOM	✓	●	–	◐	✓	◐	○	●	○	○	×	◐	✓	×	◐		
	EJB	✓	●	–	◐	✓	◐	◐	✓	○	–	×	○	✓	×	◐		
	Screen Wrappers	✓	✓	×	○	×	◐	–	✓	◐	●	✓	○	–	●	×		
	APIs	✓	✓	–	◐	✓	–	✓	✓	◐	●	✓	–	✓	×	●		

Table 6.9: Evaluation Sub-Framework that Focuses on Custom-to-Packaged-to-e-business Integration

6.3.9. Framework for Evaluating EAI Packages

Empirical evidence confirms the literature findings (Linthicum, 2000; 2001; Puschmann and Alt, 2001; Ring and Ward-Dutton, 1999) that EAI packages consist of a set of integration technologies with each EAI vendor customising its own package. This means that EAI vendors configure their products using integration technologies (e.g. adapters) to support a specific market (e.g. custom-to-packaged applications integration). For instance, one vendor may use adapters to support packaged-to-packaged integration, where another may use adapters to support custom-to-packaged applications integration. This implies that organisations may need a framework to evaluate EAI packages, and understand their capabilities before proceeding to the adoption of a solution. As a result, a framework that supports the evaluation of EAI packages may be considered as a factor that will influence EAI adoption.

The empirical data presented in Chapter 5 have shown that eight criteria were used by AUTOCORP for the evaluation of EAI packages. These criteria confirm the literature findings (Puschmann and Alt, 2001; Ring and Ward-Dutton, 1999). Nonetheless, the author puts together the eight evaluation criteria with the four integration layers in a single framework for the evaluation of EAI packages. The reason for this is that data analysis suggests taking into consideration integration layers when evaluating integration technologies. Such a framework allows decision-makers to understand which integration layers an EAI package supports as well as to realise the integration technologies that are used. The latter allows decision-makers to refer to the framework for evaluating integration technologies, and assess these technologies. This *imposes a correlation between two* evaluation frameworks. In addition, decision-makers can evaluate EAI packages using the eight criteria identified by AUTOCORP and to further clarify the differences among EAI packages. Table 6.10 presents the proposed framework for the evaluation of EAI packages. This framework was included in the conceptual model analysed in Chapter 3, however, it derives from the empirical data analysed in Chapter 5. Such a framework is a factor that influences the adoption of EAI. The framework contributes to the selection of appropriate EAI packages. It also improves IT sophistication since it contributes to understanding the capabilities of EAI packages.

EAI Vendor	EAI Product	Integration Layers: Connectivity	Transportation	Translation	Process Automation	Criteria: Integrated	Toolkit	Loosely	Tightly	Custom	Packaged	Intra-organisational	Inter-organisational
BEA Systems	Elink	Third Party	Information Broker (Message based TP Monitor)	a) Mercator (Message Broker) b) Mercator Adapters	InConcert (Process Modelling Tool)	✗	✓	✓	✓	✓	✗	✓	✗
Cross-Worlds	United Applications Architecture	Object Request Brokers (Java), APIs	InterChange Server (Java ORB)	a) InterChange Server b) Adapters	Third Party	✓	✗	✗	✓	✗	✓	✓	✓
IBM	MQ Series Integrator	a) Middleware b) Third Party	MQSeries (MOM)	a) MQSeries Integrator (Message Broker) b) Adapters	MQ Workflow (Workflow Modelling Tool)	✗	✓	✓	✗	✓	✗	✓	✗
Level 8 Systems	Enterprise Integration Template	Unified View Engine (OBDC, CORBA, COM)	Third Party	Geneva Enterprise Integrator (Message Broker)	Geneva Enterprise Integrator (Message Broker)	✗	✓	✓	✓	✗	✓	✓	✗
Mercator Software	Mercator	a) Middleware b) Third Party	Third Party	a) Mercator (Message Broker) b) Mercator Adapters	Mercator (Message Broker)	✗	✓	✓	✗	✓	SAP	✓	✓

Table 6.10: Framework for Evaluating EAI Packages

6.3.10. Support

Oliver and Romm (2000) and Stefanou (2000) identify *support* as a factor that influences the adoption of ERP systems. Support affects the introduction of enterprise application integration in organisations. This issue has been investigated and verified through the case studies of OILCORP and AUTOCORP. Both organisations have reported that: (a) vendor support and, (b) vendors global presence affect the decision for introducing an EAI solution. Therefore, influencing the adoption of EAI. This imposes a link between support factors and the framework for evaluating EAI packages.

Consultants' support is another important parameter that affects the adoption of enterprise application integration. As presented in both case studies reported in Chapter 5, consultants supported the IT department to introduce and evaluate EAI at OILCORP and AUTOCORP. In doing so, supported and influenced the decision-making process. Moreover, support factors improved IT sophistication and enhanced the organisations' knowledge regarding applications integration and EAI.

Vendors' support has a correlation with IT infrastructure since vendors provide services (e.g. maintenance) to the organisations. As reported in the case of OILCORP the: (a) close relationships between OILCORP and one of its hardware vendors and, (b) the dependence of the former on a vendors solutions (hardware), influenced the decision for purchasing EAI package from this vendor. In addition, the same vendor has provided most of the hardware required for the development of the integrated mega-datacentres. This, indicates how vendors' support may influence the decisions for adopting integrated solutions (EAI software and hardware).

6.4. Conclusions

The case for the identification of factors that affect the adoption and evaluation of EAI and the development of a model, that is translated into a frame of references has been argued, justified and presented. This chapter has concentrated on revising the conceptual model proposed in section 3.6. Modifications to the conceptual model were imposed by empirical data presented and analysed in Chapter 5. Empirical evidence suggests that apart from the factors reported in the conceptual model (see Figure 3.3) new factors should be considered when adopting and evaluating EAI. One of these new factors is internal pressure and it is reported in both case organisations as a factor that influences EAI adoption. Internal pressures can be formed as a result of: (a) operational; (b) managerial; (c) strategic; (d) technical and, (e) organisational pressures taking place within the bounds of an organisation.

Another factor that is derived by the case studies and is added to the conceptual model is the existence of an evaluation framework for the assessment of EAI packages. As explained in section 6.3.9. such a framework can be used in parallel with the framework for evaluating integration technologies to support decision making. In doing so, it contributes in reducing the confusion surrounding integration marketplace. The reason for this is that it supports better understanding regarding the applicability of each EAI package and therefore, highlights those EAI packages that are more appropriate for adoption by an organisation.

In support of this evidence a revised conceptual model has been proposed in this chapter. The revised model proposes that ten factors influence the adoption of enterprise application integration in organisations. These factors include: (a) costs; (b) barriers; (c) benefits; (d) internal pressures; (e) external pressures; (f) the level of IT sophistication; (g) the limitations of existing IT infrastructure; (h) the existence of a framework for the evaluation of integration technologies (i) a framework for evaluating EAI packages and, (j) support factors. Apart from these factors, the revised conceptual model suggests sub-factors that are related to some of these ten factors. These sub-factors include the following:

- Direct, indirect human and indirect organisational cost sub-factors that are associated with cost factor;
- Operational, managerial, strategic, technical and operational sub-factors which are related to: (a) benefits; (b) barriers and, (c) internal pressures;
- Trading partners and competitors pressures that are associated with external pressures and;
- Consultants' support, vendors' global presence and vendors' support sub-factors that are related to support factor.

All these sub-factors lead to better understanding and analysis of the factors of the revised conceptual model. Thus, they contribute to better decision-making during the process of EAI adoption and evaluation.

The novelty of the conceptual model presented in Figure 6.1, focuses on a the following:

- Based on the literature review reported in Chapter 2 there is an absence of theoretical models for EAI adoption and evaluation. Therefore, this model is one of the first attempts to explore and understand the adoption and evaluation of EAI.

- The model consists of comprehensive set of factors that influence EAI adoption and it incorporates factors identified separately in previous studies as influencing the adoption of other technologies like EDI. These factors are used for the development of a consistent model for the adoption and evaluation of enterprise application integration.

- The conceptual model can be used as a tool for decision-making to support organisations and allow researchers to apprehend and analyse EAI adoption.

- The concepts of the proposed model can be used for the adoption of inter-organisational information systems since such systems should focus on integrated technologies and infrastructures.

Chapter 7: Conclusions and Further Research

Summary

The purpose of this chapter is twofold: (a) to conclude the research carried out in this dissertation and, (b) to propose areas of further work. The chapter begins by summarising the thesis and drawing the conclusions that derived from both the literature and empirical research reported in this dissertation. Thereafter, a critical evaluation of the research process is presented. The novelty claimed in this dissertation is then summarised. The chapter ends with recommendations for further research in the area of enterprise application integration.

7.1 Research Overview and Findings

This dissertation started with an overview of the research problem in Chapter 1. It has been identified in the literature and empirically confirmed that applications integration has been a significant problem for organisations. Intra and inter-organisational systems were not developed in a co-ordinated way but have evolved as a result of the latest technological innovation. As a result, many companies have consisted of a set of complex islands of technology with diverse information formats, heterogeneous computing platforms, and various programming models. The results of this have led to incompatible systems, which have presented multiple integration problems. Previous approaches to applications integration have been proved insufficient, since they lead to complicated non-maintainable and non-cost effective solutions. Enterprise resource planning systems were then proposed as integrated suites and promised applications integration. However, ERP systems have failed to achieve integration, as they were not designed to collaborate with other existing or new applications. Hence, organisations seek new ways to achieve integration, since integration has a significant role in modern business environments. The implication of this is that organisations require more flexible IT infrastructures that will allow them to easily adapt to the changing business arena, as well as to gain a competitive advantage. In this context, enterprise application integration has emerged as a technology to effectively integrate intra and inter-organisational systems. In doing so, EAI incorporates functionality from disparate applications by combining diverse technologies such as adapters and message brokers. Chapter 1 then, states the aim of this research, which is to *evaluate the adoption of EAI in multinational organisations. In doing so, resulting in the development of a frame of references which translates into a model that can be used to support decision-making.* Thereafter, Chapter 1 provides a general overview to the dissertation outline.

In an attempt to meet the aim of this dissertation, Chapter 2 (background theory) started with a literature review on the motivations to EAI adoption. Much of these motivations are empirically confirmed and then reflected in Chapters 5 and 6. In investigating more the adoption of EAI, Chapter 2 reviews the normative literature and seeks models and factors that influence EAI adoption. To this end, it has identified that there is an absence of theoretical models that deal with EAI adoption, since enterprise application integration is relatively new research area. As a result, the author has drawn parallels between EAI and another integration technology such as EDI and, summarised the factors that influence the adoption of EDI technology. In doing so, supporting the investigation of the factors that may affect the introduction of EAI in organisations. This is based on EDI and EAI support systems following similar integration concepts. Chapter 2 then discussing the results of the literature review on EAI. Since, there is much confusion regarding EAI terminology and functionality, the author critically evaluates EAI terminology (see Appendix A) and clarifies the functionality of EAI. In support of this, a novel taxonomy for classifying types of application integration is proposed. Such taxonomy allows researchers and decision-makers to apprehend the whole range of EAI functionality and thus, improves IT sophistication. Another contribution derived from this dissertation and reported in Chapter 2, deals with the identification of system types that are integrated. Based on an extensive literature review, seven system types (e.g. custom-to-ebusiness applications integration) have been identified and are empirically confirmed (see Chapter 5). This also contributes to an extension of EAI knowledge, since there has been a debate regarding the system types that EAI integrates (see Appendix A). In identifying system types, the dissertation results in reducing much of the confusion surrounding EAI. Thereafter, a literature review of EAI costs, barriers and benefits is presented. The reason for this is twofold: firstly these parameters provide a better understanding of EAI and secondly, benefits, barriers and costs are reported as factors that influence EDI adoption. The latter indicates that these factors may also influence EAI adoption. The author categorises benefits, barriers and costs using appropriate classifications from the literature. In doing so, Chapter 2 makes further contribution to the EAI literature since it enhances knowledge by proposing classifications for EAI benefits, barriers and costs. The proposed classifications can be used to support EAI analysis and evaluation. In reviewing EAI barriers and thus justifying the thesis, an important research issue has emerged, which indicated that the technological confusion surrounding EAI is an obstacle to its adoption.

Chapter 3 (focal theory) has concentrated on investigating the research issues that derived from Chapter 2. In doing so, Chapter 3 proposed: (a) a framework for evaluating integration technologies and, (b) a conceptual model for EAI adoption and evaluation. Initially, a technical approach to EAI has been presented to provide a deeper understanding of this technology. Important parameters that influence the evaluation of integration technologies have also been highlighted. These parameters deal with: (a) integration layers; (b) applications elements and, (c) integration requirements. The author has argued, identified and empirically verified that these parameters can consist of three categories of evaluation criteria for the assessment of integration technologies. In addition to these categories, the system types that are integrated through EAI and have also been identified in Chapter 2; consist the fourth category of evaluation criteria. All these categories of criteria result in a novel comprehensive framework for the evaluation of integration technologies. To this end, a summary of integration technologies has been presented to support readers in understanding the evaluation of technology process. Thereafter, evaluation criteria has identified for the assessment of integration technologies (see Table 3.3). Thereafter, the novel evaluation framework has been proposed (see Table 3.4) to reduce the confusion surrounding the integration area, and to support the selection of appropriate technologies. An evaluation of integration technologies has then been presented using both the proposed framework and literature findings. In doing so, the author demonstrated that such a framework supports decision-making process. The presented framework consists an influencing factor for EAI adoption. In investigating more factors that affect the adoption of EAI, a novel conceptual model was identified (see Figure 3.3). This meets the aim of this dissertation reported in Chapter 1. The conceptual model was then empirically examined and modified in Chapter 5 and 6 of this dissertation.

To undertake the research that focuses on the issues identified in Chapters 2 and 3, a research methodology was developed and adopted. Justification for the research methods is stated within Chapter 4 (data theory). The inherent problems within the various research philosophies are stated and the suitability to this research outlined. The research strategies existing within the IS field are also described and discussed within this chapter.

The research issues that have been identified in previous chapters of this dissertation were investigated through the use of case studies in two multinational organisations. These issues dealt with the factors that influence the adoption of enterprise application integration. Chapter 5 (data theory) then presented and analysed empirical evidences and offered an empirical analysis of different case study perspectives. In doing so, it described human and organisational perceptions during the adoption process of application integration. Empirical evidences derived from the cases of OILCORP and AUTOCORP also confirmed much of the issues identified in Chapters 2 and 3. However, a number of additions for the conceptual model as well as the evaluation framework proposed and based on empirical data.

Empirical evidences that derived from the cases of OILCORP and AUTOCORP have indicated a number of modifications to the conceptual model and the framework for integration technologies evaluation. These findings have been considered in Chapter 6 (novel contribution) and resulted in the revision of both the conceptual model (see Figure 6.1) and the evaluation framework (see Table 6.2). The revised conceptual model supports that ten factors influence the adoption and the evaluation of EAI in organisations. These factors deal with:

- costs;
- barriers;
- benefits;
- internal pressures;
- external pressures;
- the level of IT sophistication;
- the limitations of existing IT infrastructure;
- the existence of a framework for the evaluation of integration technologies
- a framework for evaluating EAI packages; and,
- support factors.

The conceptual model and the proposed frameworks for EAI packages and integration technologies evaluation can therefore, be used as frames of references when organisations taking their decisions for EAI adoption. In doing so, the author has achieved the aim of this dissertation as identified in section 1.5.

The main findings derived from the work presented in this dissertation are presented below:

- A review of normative literature suggests that there is an absence of theoretical models that describe the adoption and evaluation of enterprise application integration. The reason for this is attributed to that EAI is a new research area with many topics remaining under investigation.

- Literature review on integration area designates that the technological confusion surrounding integration marketplace is an obstacle to EAI adoption. The reason for this is that many integration technologies and EAI packages exist in marketplace but there is no single product overcoming all integration problems. Therefore there is a need for evaluation frameworks that will support the selection of appropriate technologies and EAI packages.

- The author attempts to address these voids in literature by proposing and empirically establish a novel conceptual model for the adoption and evaluation of EAI. The proposed model is based on a consistent set of ten influential factors for EAI adoption namely: (a) costs; (b) barriers; (c) benefits; (d) internal pressures; (e) external pressures; (f) IT infrastructure; (g) IT sophistication; (h) support; (i) the existence of a framework for the evaluation of integration technologies and, (j) a framework for the assessment of EAI packages.

- The conceptual model can be used as a tool for decision-making to support organisations and allow researchers to apprehend and analyse EAI adoption. The concepts of the proposed model can be used for the adoption of inter-organisational information systems since such systems should focus on integrated technologies and infrastructures.

- It is empirically verified through the case studies of OILCORP and AUTOCORP that the two evaluation frameworks identified in the proposed model can be used for the assessment, selection and adoption of appropriate integration technologies and EAI packages. Empirical evidences indicate that the use of the two proposed frameworks increases IT sophistication and supports decision making for EAI adoption.

- The author classifies EAI benefits using the model proposed by Shang and Seddon (2000). This model divides benefits that derived from integrated IT infrastructures into: (a) operational; (b) managerial; (c) technical; (d) strategic and, (e) organisational. The author claims and empirically verifies that the same model can be used for the classification of EAI barriers. In addition, the empirical evidences led the author to classify costs into (a) direct; (b) indirect human and, (c) indirect organisational costs. In doing so, continuing the application of the cost model proposed by Hochstrasser (1992).

- Based on extensive literature review, the author attempts to clarify the confusion regarding the EAI terminology and applicability. In doing so, the research presented in this thesis indicates that EAI can be used for the adoption of both intra-organisational and inter-organisational systems. In addition the system types that are integrated using EAI are identified and classified in seven types including: (a) custom-to-custom; (b) custom-to-packaged; (c) custom-to-e-business; (d) packaged-to-packaged; (e) packaged-to-e-business; (f) e-business-to-e-business and, (g) custom-to-packaged-ebusiness.

7.2 Research Limitations

In developing a model for the adoption and evaluation of enterprise application integration that can be used as a frame of references for decision-makers, there was a need for a robust research methodology. Such a methodology could be used as a framework for developing other application specific models for the adoption of technology.

As described in Chapter 4, the use of qualitative data gathering methods were justified for gathering the necessary data. The reason for this is that such methods allow generalisation of *soft, rich* contextual data, which is associated with human and organisational issues. Despite its strengths, qualitative research methods do have inherent weaknesses, with a number being encountered during reported research process. In conducting this research, the collection and analysis of qualitative data has proved time consuming and demanding.

Nonetheless, the relative difficulty of analysing this data did not invalidate any conclusions drawn, since multiple case studies analysis was applied to data obtained.

Moreover, the author has acknowledged a number of additional issues regarding the use of qualitative research methods. Firstly, the inability of the researcher to interpret events from the subject point of view, is questioned, without some degree of bias. However, to try and address this the author uses a multi-method approach [data triangulation] to data gathering. Although, the author does consider that there will always be elements of bias inherent in qualitative data analysis, due to its subjective nature. Secondly, the relationship between theory and research might be considered weak and unstructured, as qualitative approaches may be criticised for not instilling theoretical elements. However, in the case of this research, the author sought to partially address this concern through developing a conceptual model proposing factors that influence EAI adoption, and building a framework for evaluating integration technologies. Although retrospectively, the author considers that a lack of structure and theory can actually add the diversity and 'richness' of qualitative data gathering. As a result, the appropriateness of grounded theory is now appreciated and acknowledged as a suitable research methodology for investigating EAI adoption.

Finally, there is much concern regarding the extent that qualitative research can be generalised beyond the confines of the inquiry, as the sample of companies are often relatively few. Even though the number of companies used during this study was two to extent this inquiry further would not have increased its external validity. Indeed, qualitative case study research does not offer the pretence of replication, as controlling the research setting destroys the interaction of variables, and therefore, affects the underline philosophy of interpretivism. In re-assuring sceptics of interpretivism, the study was conducted within a structured methodology, and guided by theoretical concepts and models, with a number of data gathering methods and processes having been used. However, the methodology presented in Chapter 4 was developed as it was considered safer to identify and investigate independent variables following a review of literature. Having now evaluated the research process, such concern needed not of been considered important, as a grounded theory approach may also have been suitable, and yet, still provided 'freedom' and scope for: (a) discovery and theory building and, (b) discovery, theory building and testing.

7.3 Statement of Contribution and Research Novelty

The individual elements of the contributions made by this work stem from different components in this dissertation. From the contextual information provided in Chapters 1, 2 and 3, to the research methodology reported in Chapter 4, through the design and the conduct of the case studies reported in Chapters 4 and 5 and finally the empirical analysis of the cases and the development of conceptual model presented in Chapter 5 and 6. The work presented in this dissertation has made novel contribution to the area of applications integration and has extended the boundaries of knowledge. The following is a review of what is sustained to be the main contribution and research novelty of the thesis.

7.3.1 A Novel Model for EAI Adoption and Evaluation

The most important contribution of this dissertation is the development of a comprehensive novel model for EAI adoption and evaluation (see Figure 6.1). As described earlier in sections 2.2 and 2.9, there is a lack of theoretical models describing EAI adoption and evaluation. In addressing this void in the literature, section 3.6 proposes a conceptual model for EAI adoption. This model is empirically investigated and analysed in Chapter 5. Empirical evidences derived from case studies have resulted in a revision of the model (see Figure 6.1) and have led to the final model for EAI adoption being presented in section 6.3.

The model makes novel contribution at two levels. Firstly, at the conceptual level, the model incorporates factors identified in previous studies as influencing the adoption of integration technologies like EDI. The author extents these works and adapts them to the application integration area by combing factors discussed in the normative literature. In addition, influencing factors that derived from empirical evidences are also incorporated in the proposed model. In doing so, resulting in the development of a consistent model for the adoption and evaluation of application integration. Secondly, the concepts of the proposed model can be used as a guide for the adoption of inter-organisational information systems. A novel aspect of the proposed model is that it introduces frameworks for the evaluation of integration technologies and EAI packages as factors, which influences the adoption of application integration. The frameworks themselves increase IT sophistication and support understanding and decision-making.

7.3.2 A Novel Framework for Evaluating Integration Technologies

Another important contribution made in the thesis, deals with the proposition of a novel evaluation framework (see Table 6.2), which supports the assessment of integration technologies. Section 2.6 has identified that an important barrier to EAI adoption is the confusion surrounding integration area. These issues have been empirically verified in sections 5.2.6 and 5.3.6. In addressing this barrier a framework for evaluating integration technologies has been proposed in Chapter 3 and confirmed in Chapter 5. In addition, empirical evidences have indicated revisions to the framework. Section 6.3.6 has revised the framework by incorporating the additional criteria that derived from the empirical case studies (see Table 6.2).

The evaluation framework is based on criteria that are derived from a comprehensive literature review and empirical data. Evaluation criteria are divided into four categories namely: (a) integration requirements; (b) the application elements that are integrated; (c) the integration layers and, (d) the classification of system types that are integrated. This framework can be used as a frame of references to highlight possible combinations of technologies that can support the integration of an IT infrastructure. The proposed evaluation framework clarifies the differences among technologies and will support integrators during the selection of an appropriate permutation of integration technologies. In addition, such a framework can be used as a tool that allows organisations to develop their own bespoke integration solution. On a technical level, the evaluation framework contributes towards a better understanding of the capabilities of each technology, and highlights combinations of integration solutions. The framework highlights possible permutations of technologies that can address the integration of IS. Such a framework can be used as a decision making tool by IT departments or business analysts when taking decisions to integrate enterprise and cross enterprise applications.

7.3.3 A Novel Framework for Evaluating EAI Packages

Empirical data revealed from the reported case studies confirmed literature findings presented in Chapters 2 and 3. Case data indicated that there is a need to evaluate EAI packages before selecting one. The need to better analyse and evaluate EAI packages have become crucial for organisations, since literature and practice have indicated that there is no single EAI package solving all integration problems. In support of this finding the author has proposed a novel framework to evaluate EAI packages (see Table 6.10).

Similarly to the framework for evaluating integration technologies, this one uses criteria that have been identified in the literature or elicited from empirical data. The combination of criteria makes the framework novel since they result in a comprehensive set of criteria that efficiently support decision-making. Evaluation criteria include the integration layers (e.g. connectivity, process automation layer) as well as attributes of EAI packages. Such attributes focus on the capabilities of EAI packages to: (a) support intra and inter-organisational integration; (b) support loose and tight applications coupling; (c) support custom and packaged systems integration and, (d) provide individual or toolkit applications. This framework as well as the one for the evaluation of integration technologies contributes to decision-making and reduces the confusion surrounding the integration marketplace.

7.3.3 Classifications of EAI Barriers, Benefits and Costs

A further contribution deals with the preposition of classifications of EAI barriers, benefits and costs. In Chapter 2, barriers and benefits classifications were based on the model proposed by Shang and Seddon (2000), which was proposed for the categorisation of ERP benefits. In section 2.7.1, EAI costs are classified following a typical direct and indirect (human and organisational) classification, similar to the one reported by Irani (1998). The novelty claimed is that such classifications can be adapted and followed in the case of EAI. In doing so, the author expands existing knowledge on EAI, since there is an absence of classifications of EAI benefits, barriers and costs. Therefore, supports decision-makers and researchers to better understand the impact of such technology and thus, supporting robust evaluation.

7.4 Recommendations for Further Work

The identification and the development of a model for EAI adoption and evaluation has established those issues that appear crucial within the two multinational organisations studied. To refine such a model as well as the two evaluation frameworks proposed in this dissertation may be considered to further substantiate the research presented. Therefore, the author suggests transforming the proposed model and frameworks into a large-scale survey questionnaire, rather than continuing with an interpretivist epistemology. Clearly, this approach would not have been possible previously, since the model and the frameworks did not exist. A large-scale survey will offer the opportunity to establish generic significance to the issues related to the proposed frameworks and model (e.g. evaluation criteria, factors). In surveying a representative sample of organisations the criteria and factors related to the proposed frameworks and model can be better verified and understood.

Another important research proposition is to establish whether the model can be used by Small and Medium size Enterprises (SMEs) or only by large organisations. As reported in Chapter 5, both organisations been studied are large multinational enterprises. Therefore, an interesting area for further research could be to investigate the adoption of EAI by SMEs. It is proposed that this can be achieved using a similar methodology to that presented in Chapter 4. Moreover, the proposed frameworks for the evaluation of integration technologies and EAI packages can be examined to establish whether they are applicable in SMEs.

Another recommendation is to further study the factors that influence the adoption of EAI. Such a study may focus on the importance of each factor, and identify if these factors have the same influence or some of them more are more significant. Such research would clarify whether some factors are more important to others. This will improve analysis in this area and contribute in better decision-making.

References

Al-Mashari, M. 2000. *'Constructs of Process Change Management in ERP Content: A Focus on SAP R/3'*, Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 977-980.

Al-Mashari, M. and Zairi, M. 2000. 'Supply-Chain Re-engineering using the Enterprise Resource Planning (ERP) Systems: An analysis of a SAP R/3 Implementation Case', *International Journal of Physical Distribution and Logistics Management,* **30**(3/4): 296-313.

AMR Research 1999. *'AMR Research Predicts ERP Market Will Reach $66.6 billion by 2003'.*

Andrew, D. 1998. *'Component-Based Development',* Butler Group Limited, Yorkshire, UK.

Angeli, A., Streit, U. and Gonfalonieri, R. 2000. *'The SAP R/3 Guide to EDI and Interfaces',* Friedrick Vieweg & Son/Morgan Kaufmann Publishers, Berlin, Germany.

Anonymous 1999a. *'Catawba Memorial Hospital - Hickory, North Carolina',* International Data Corporation, http://www.seebeyond.com/customers/roicasestudies.html

Anonymous 1999b. *'National Power - London, UK',* *International Data Corporation,* http://www.seebeyond.com/customers/roicasestudies.html

Anonymous 1999c. *'PETsMART - Phoenix, Arizona',* International Data Corporation, http://www.seebeyond.com/customers/roicasestudies.html

Anonymous 2001. *'The Case of Amazon.com',* http://www.isss-awareness.cenorm.be/Case_Studies/Amazon_case_study.htm

Bakos, Y. 1998. 'The Emerging Role of Electronic Marketplaces on the Internet', *Communications of the ACM,* **41**(8): 35-42.

Barry, D. 1996. *'Data Warehouse: From Architecture to Implementation',* Addison-Wesley, Boston, Massachusetts, USA.

Becker, J. D., Farris, T. and Osborn, P. 1998. *'Electronic Commerce and Rapid Delivery: The Missing 'Logistical' Link',* Proceedings of 4th Americas Conference on Information Systems, Baltimore, Maryland, USA, pp. 272-274.

Benbasat, I., Goldstein, D. K. and Mead, M. 1987. 'The Case Research Strategy in Studies of Information Systems', *MIS Quarterly,* **11**(3): 369-386.

Bennett, K. 1995. 'Legacy Systems: Coping with Success', *IEEE Software,* **12**(1): 19-23.

Bernroider, E. 1999. *'Decision Making for ERP Investments from the Perspective of Organizational Impact - Preliminary Results from an Empirical Study'*, Proceedings of 5th Conference on Information Systems, AMCIS 1999, Milwaukee, Wisconsin, USA, pp. 773-775.

Bernstein, P. 1990. 'Transaction Processing Monitors', *Communications of the ACM*, **33**(11): 75-86.

Bernstein, P. 1996. 'Middleware: A Model for Distributed System Services', *Communications of the ACM*, **39**(2): 86-98.

Bernus, P., Nemes, L. and Williams, T. 1996. *'Architectures for Enterprise Integration'*, Chapman & Hall, London, UK.

Bhatt, G. and Emdad, A. 2001. 'An Analysis of the Virtual Value Chain in Electronic Commerce', *Logistics Information Management*, **14**(1/2): 78-84.

Birrell, A. and Nelson, B. 1984. 'Implementing Remote Procedure Calls', *ACM Transactions on Computer Systems*, **3**(1): 39-59.

Blaxter, L., Hughes, C. and Tight, M. 1996. *'How to Research'*, Open University Press, Trowbridge, UK.

Bonoma, T. 1985. 'Case Research in Marketing: Opportunities, Problems and a Process', *Journal of Marketing Research*, **22**:199-208.

Brancheau, J. C., Janz, B. D. and Wetherbe, J. C. 1996. 'Key Issues in Information Systems Management: 1994-95 SIM Delphi Results', *MIS Quarterly*, *20(2): 225-242.*

Brancheau, J. C. and Wetherbe, J. C. 1990. 'The Adoption of Spreadsheet Software: Testing Innovation Diffusion Theory in the Context of End-User Computing', *Information Systems Research*, **1**(2): 115-143.

Britton, C. and Doake, J. 1996. *'Software System Development: A Gentle Introduction'*, Alfred Waller, London, UK.

Brodie, M. and Stonebraker, M. 1995. *'Migrating Legacy Systems'*, Morgan Kaufmann Publishers, San Francisco, USA.

Brown, C., Vessey, I. and Powell, A. 2000. *'The ERP Purchase Decision: Influential Business IT Factors'*, Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 1029-1032.

Brown, L. 2000a. *'Integration Models: Templates for Business Transformation'*, SAMS Publishing, USA.

Brown, T. 1999. *'The Ptolemy Error'*, *EAI Journal*, September/October 1999, pp. 64

Butler Group 1998. 'Legacy Wrapping and Application Integration', *Component-Based Development Journal,* **August 1998:**2-7.

Bytheway, A. and Dhillon, G. 1996. 'Significance of Partnerships in the Management of Interorganisational Systems', *International Journal of Information Management,* **16**(5): 369-380.

Camp, R. 1991. *'Benchmarking: The Search for Industry Best Practices that Lead to Superior Performance',* ASQC Quality Press, Milwaukee, WI, USA.

Camp, R. 1995. *'Business Process Benchmarking: Finding and Implementing Best Practices',* ASQC Quality Press, Milwaukee, WI, USA.

Carrier, L. 1999. 'Managing at Light Speed', *IEEE Computer,* **July:**107-109.

Cavaye, A. 1996. 'Case Study Research: A Multi-Faceted Research Approach for IS', *Information Systems Journal,* **6**(3): 227-242.

Choudhury, V. 1997. 'Strategic Choices in the Development of Interorganisational Information Systems', *Information Systems Research,* **8**(1): 1-24.

Chung, S. and Snyder, C. 2000. 'ERP Adoption: A Technological Evolution Approach', *International Journal of Agile Management Systems,* **2**(1): 24-32.

Chwelos, P., Benbasat, I. and Dexter, A. 2001. 'Research *Report: Empirical Test of an EDI* Adoption Model', *Information Systems Research,* **12**(3): 304-321.

Cingil, I., Dogac, A. and Azgin, A. 2000. 'A Broader Approach to Personalization', *Communications of the ACM,* **43**(8): 136-141.

Coad, P. and Yourdon, E. 1990. *'Object-Oriented Analysis',* Prentice-Hall Inc., New Jersey, USA.

Connall, D. and Burns, D. 1993. 'Reverse Engineering: Getting a Grip on Legacy Systems', *Data Management Review,* **24**(7).

Davenport, T. 1998. 'Putting the Enterprise into the Enterprise System', *Harvard Business Review,* July-August 1998: 121-131.

Denzin, N. and Lincoln, Y. 1994. *'Handbook of Qualitative Research',* SAGE Publications, London, UK.

Denzin, N. K. and Lincoln, Y. S. 1998. *'Collecting and Interpreting Qualitative Materials',* SAGE Publications, Thousand Oaks, California, USA.

Denzin, N. Y. and Lincoln, Y. 2000. 'The Discipline and Practice of Qualitative Research' In Handbook of Qualitative Research' (Ed, Lincoln, Y.) SAGE Publications, Thousand Oaks, California, USA, pp. 1-28.

Denzin, N. Y. K. 1978. *'The Research Act: A Theoretical Introduction to Sociological Methods',* McGraw Hill, New York, USA.

Dong, L. 2000. *'A Model for Enterprise Systems Implementation: Top Management Influences On Implementation Effectiveness',* Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 1045-1049.

Doukidis, G., Themistocleous, M., Drakos, W. and Papazafeiropoulou, A. 1998b. *'Electronic Commerce',* New Technology Publications, Athens, Greece.

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

Dyer, W. G., Wilkins, A. L. and Eisenhardt, K. M. 1991. 'Better Stories, not Better Constructs, to Generate Better Theory', *Academy of Management Review,* **16**(3): 613-627.

Eck, J. and Marchetti, N. 2000. *'Combing e-Commerce & EAI', EAI Journal,* January 2000, pp. 42-43

Edwards, P. and Newing, R. 2000. *'Application Integration for e-Business',* Business Intelligence 2000, London, UK.

Eisenhardt, K. M. 1989. 'Building Theories from Case Study Research', *Academy of Management Review,* **14**(4): 532-550.

Emmelhainz, M. 1993. *'EDI: A Total Management Guide',* Van Nostrand Reinhold, New York, USA.

Flick, U. 1998. *'An Introduction to Qualitative Research: Theory, Method and Applications',* SAGE Publications, London, UK.

Forrester Research 1998. *'Commerce Technologies Strategies',* Forrester Research.

Frey, J. H. and Fontana, A. 1991. 'The Group Interview in Social Research', *Social Science Journal,* **28**:175-187.

Gable, G. 1994. 'Integrating Case Study and Survey Research Methods: An Example in Information Systems', *European Journal of Information Systems,* **3**(2): 112-126.

Galal, G. 1996. *'An Interpretive Approach to Information Systems Engineering using the Grounded Systems Engineering Methodology', PhD Thesis, Department of Information Systems and Computing,* Brunel University, London, UK.

Galal, G. 2001. 'From Contexts to Constructs: The use of Grounded Theory in Operationalising Contingent Process Models', *European Journal of Information Systems,* **10**(1): 2-14.

Galal, G., McDonnell, J. and Paul, R. 2001. *'The Utility of Qualitative Modeling in Architectural Evaluation'*, Proceedings of Seventh Americas Conference on Information Systems, AMCIS 2000, Boston, Massachusetts, USA, pp. 1392-1395.

Galliers, R. D. 1992. *'Chosing Information Systems Research Approaches',* Blackwell Scientific, Oxford, UK.

Galliers, R. D. 1994. 'Points in Understanding Information Systems Research', *Systemist,* **16**(1): 32-40.

Gibson, N., Holland, C. and Light, B. 1999. *'Enterprise Resource Planning: A Business Approach to Systems Development'*, Proceedings of 32nd Hawaii International Conference on System Sciences, Hawaii, USA, (CD Proceedings).

Gilbert, A. and Sweat, J. 1999. *'ERP Vendors Get Serious About Application Integration',* Information Week, http://www.informationweek.com/759/eai.htm

Glaser, B. and Strauss, A. 1967. *'The Discovery of Grounded Theory - Strategies for Qualitative Research',* Weindenfeld and Nicolson, London, UK.

Glass, R. and Vessey, I. 1999. *'Enterprise Resource Planning Systems: Can They Handle the Enhancement Changes Most Enterprises Require?'* Proceedings of First International Workshop on Enterprise Management and Enterprise Resource Planning Systems: Methods, Tools and Architectures (EMRPS'99), Venice, Italy.

Graham, G. and Hardaker, G. 2000. 'Supply-Chain Management across the Internet', *International Journal of Physical Distribution and Logistics Management,* **30**(3/4): 286-295.

Grimshaw, D. 1999. *'The Potential for Spatial Data Integration: A Case Study of Geographical Knowledge in a Small Business'*, Proceedings of 5th Conference on Information Systems, AMCIS 1999, Milwaukee, Wisconsin, USA, pp. 271-273.

Grimson, J., Grimson, W. and Hasselbring, W. 2000. 'The SI Challenge in the Health Care', *Communications of the ACM,* **43**(6): 49-54.

Hammer, M. and Champy, J. 1993. *'Reengineering the Corporation: A Manifesto for Business Revolution',* HarperCollins, New York, USA.

Handfield, R. and Nichols, V. 1999. *'Introduction to Supply Chain Management',* Prentice Hall, Englewood Cliffs, New Jersey, USA.

Hasselbring, W. 2000. 'Information System Integration', *Communications of the ACM,* **43**(6): 33-38.

Heckman, R. 1998. 'Planning to Solve the "Skills Problems" in the Virtual Information Management Organisation', *International Journal of Information Management,* **18**(1): 3-16.

Helm, R. 1999. *'Extending EAI Beyond the Enterprise', EAI Journal,* www.eaijournal.com

Henwood, K. and Pidgeon, N. 1995. 'Grounded Theory and Psychological Research', *The Psychologist,* **March 1995:**115-118.

Herriot, R. E. and Firestone, W. A. 1983. 'Multi-site Qualitative Policy Research: Optimising Description and Generalisability', *Educational Researcher,* **12:**14-19.

Hicks, D. 1997. 'The Managers' Guide to Supply Chain and Logistics Problem-Solving Tools and Techniques', *IIE Solutions,* **29**(10): 24-29.

Hochstrasser, B. 1992. *'Justifying IT Investments'*, Proceedings of Advanced Information Systems Conference; The New Technologies in the Today's Business Environment, UK, pp. 17-28.

Holland, C. and Light, B. 1999. *'Global Enterprise Resource Planning Implementation'*, Proceedings of 32nd Hawaii International Conference on System Sciences, Hawaii, USA, [CD Proceedings].

Holland, C., Light, B. and Gibson, N. 1999a. *'A Critical Success Factors Model for Enterprise Resource Planning Implementation'*, Proceedings of 7th European Conference on Information Systems, Copenhagen, Denmark, pp. 273-287.

Holland, C., Light, B. and Kawalek, P. 1999b. *'Beyond Enterprise Resource Planning Projects: Innovative Strategies for Competitive Advantage'*, Proceedings of 7th European Conference on Information Systems - ECIS'99, Copenhagen, Denmark, pp. 288-301.

Hooft, F. and Stegwee, R. 2001. 'e-business *Strategy: How to Benefit from a Hype', Logistics Information Management,* **14**(1/2): 44-53.

Hoole, S. 1999. *'Bevitalized Applications at Nu Skin Enterprises', EAI Journal,* September/October 1999 www.eaijournal.com

Hu, P. J., Chau, P. and Sheng, O. 2000. *'Investigation of Factors Affecting Healthcare Organization's Adoption of Telemedicine Technology'*, Proceedings of 33rd Hawaii International Conference on System Sciences, Hawaii, USA, [CD Proceedings].

Iacovou, C., Benbasat, I. and Dexter, A. 1995. 'Electronic Data Interchange and Small Organizations : Adoption and Impact of Technology', *MIS Quarterly,* **19**(4): 465-485.

IBM 1999. *'Package Enabled Business Transformation - PEBT',* http://houns54.clearlake.ibm.com/solutions/erp/erppub.nsf/SearchResults/8625680800665E228625670700056FF3#enabled

Irani, Z. 1998. *'Investment Justification of Information Systems: A Focus on the Evaluation of MRPII', PhD Thesis, Department of Manufacturing and Engineering,* Brunel University, London, UK.

Irani, Z., Ezingeard, J.-N. and Grieve, R. J. 1997. 'Integrating the Costs of an IT/IS Infrastructure into the Investment Decision Making Process', *The International Journal of Technological Innovation and Entrepreneurship (Technovation),* **17**(11/12): 637-647.

Irani, Z., Ezingeard, J.N. and Grieve, R. J. 1998. 'Costing the True Costs of IT/IS Investments: A Focus During Management Decision Making', *Logistics Information Management,* **11**(1): 38-43.

Irani, Z., Ezingeard, J.N. Grieve, R. J. and Race, P. 1999. 'A Case Study Approach to Carrying Out Information Systems Research: A critique', *International Journal of Computer Applications in Technology,* **12**(2): 90-101.

Irani, Z. and Love, P. 2001. 'The Propagation of Technology Management Taxonomies for Evaluating Investments in Information Systems', *Journal of Management Information Systems,* **17**(3): 161-177.

Irani, Z., Sharif, A. and Love, P. 2001. 'Transforming Failure into Success through Organisational Learning: An Analysis of a Manufacturing Information System', *European Journal of Information Systems,* **10**:55-66.

Isakowitz, T., Bieber, M. and Vitali, F. 1998. 'Web Information Systems Introduction', *Communications of the ACM,* **41**(7): 78-80.

Janesick, V. 2000. 'The Choreography of Qualitative Research Design' *In Handbook of Qualitative Research* (Ed, Lincoln, Y. S.) SAGE Publications, Thousand Oaks, California, USA, pp. 379-399.

Jayaram, J., Vickery, S. and Droge, C. 2000. 'The Effects of Information System Infrastructure and the Process Improvement on Supply-Chain Performance', *International Journal of Physical Distribution and Logistics Management,* **30**(3/4): 314-330.

Jick, T. D. 1979. 'Mixing Qualitative and Quantitative Methods: Triangulation in Accumulation', *Administrative Science Quarterly,* **24**:602-611.

Kalakota, R. 2000. 'Inter-Enterprise Fusion - The Future of Supply Chains', *EAI Journal,* **May 2000**: 72-76.

Kalakota, R. and Robinson, M. 1999. *'e-Business: Roadmap for Success',* Addison-Wesley, Boston, Massachusetts, USA.

Kalakota, R. and Robinson, M. 2001. *'e-Business 2.0: Roadmap for Success',* Addison-Wesley, Boston, Massachusetts, USA.

Kaplan, B. and Maxwell, J. A. 1994. 'Qualitative Research Methods for Evaluating Computer Information Systems' *In Evaluating Health Care Information Systems: Methods and Applications* (Ed, Jay, S. J.) SAGE Publications, Thousand Oaks, California, USA, pp. 45-68.

Kaufman, F. 1996. 'Data Systems that Cross Company Boundaries', *Harvard Business Review,* **January-February.**

Kelley, H., Compeau, D. and Higgins, C. 1999. *'Attribution Analysis of Computer Self-Efficacy',* Proceedings of 5th Conference on Information Systems, AMCIS 1999, Milwaukee, Wisconsin, USA, pp. 782-784.

Kelly, S., Holland, C. and Light, B. 1999. *'Enterprise Resource Planning: A business Approach to Systems Development',* Proceedings of 5th Conference on Information Systems, AMCIS 1999, Milwaukee, Wisconsin, USA, pp. 785-787.

Kim, K. K. and Umanath, N. 1999. *'An Empirical Investigation of Electronic Integration in a Supply Chain Relationship',* Proceedings of 20th International Conference on Information Systems, Charlotte USA, [CD Proceedings].

Klasell, T. and Dudgeon, S. 1998. *'Enterprise Application Integration',* Dain Rauscher Wessels, New York, USA.

Klein, K. and Myers, M. 1999. 'A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems', *MIS Quarterly,* **23**(1): 67-94.

Knoll, K. and Jarvenpaa, S. 1994. *'Information Technology Alignment or "Fit" in Highly Turbulent Environments: the Concept of Flexibility',* Proceedings of Proceedings of the 1994 Computer Personnel Research Conference on Reinventing IS: Managing Information Technology in Changing Organizations, Alexandria, Virginia, USA.

Korson, T. and McGrecor, J. 1990. 'Understanding Object-Oriented: A Unifying Paradigm', *Communications of the ACM,* **33**(9): 40-60.

Korzeniowsky, P. 2000. *'EAI Tools Ease ERP Integration Woes', EAI Journal,* http://www.eaijournal.com/ERPIntegration/EAITools.asp

Krueger, C. 1992. 'Software Reuse', *ACM Computer Surveys,* **24**(2): 131-183.

Lambert, D. M., Cooper, M. C. and Pagh, J. D. 1998. 'Supply Chain Management: Implementation Issues and Research Opportunities', *International Journal of Logistics Management,* **9**(2): 1-19.

Langan, G. 2000. *'10 Thinks you Must Know about e-Business', EAI Journal,* January 2000, pp. 24-26

Lee, A. A. 1991. 'Integrating Positivist and Interpretive Approaches to Organisational Research', *Organisation Science,* **2**(4): 342-365.

Lee, H. 1998. 'Do Electronic Marketplaces Lower the Price of Goods?' *Communications of the ACM,* **41**(1): 73-80.

Lesley, R. 2000. *'Integration Apocalypse?' EAI Journal,* www.eaijournal.com

Ling, C. Y. 2001. *'Model of Factor Influences on Electronic Commerce Adoption and Diffusion in Small and Medium Sized Enterprises',* Proceedings of ECIS 2001, Bled, Slovenia, PhD Consortium.

Linthicum, D. 1999a. *'Enterprise Application Integration',* Addison-Wesley, Massachusetts, USA.

Linthicum, D. 1999b. *'Enterprise Application Integration from the Ground Up',* Software Development Magazine, http://sdmagazine.com/breakrm/features/s994f2.shtml

Linthicum, D. 2000a. *'B2B Application Integration',* Addison-Wesley, Massachusetts, USA.

Linthicum, D. 2000b. *'Mercator: eBusiness Integration Software in an SAP Environment',* Mercator Software, Wilton, Connecticut, USA.

Linthicum, D. 2001. *'Mercator: Next Generation Application Integration',* Mercator Software, Wilton, Connecticut, USA.

Lloyd, A., Dewar, R. and Pooley, R. 1999. 'Legacy Information Systems and Business Process Change: A patterns Perspective', *Communications of the Association for Information Systems,* **2**(Article 24): 1-40.

Lohse, G. and Spiller, P. 1998. 'Electronic Shopping', *Communications of the ACM,* **41**(7): 81-88.

Loinsky, S. 1995. *'Enterprise-Wide Software Solutions: Integration Strategies and Practices',* Addison-Wesley, Boston, Massachusetts, USA.

Longbottom, D. 2000. 'Benchmarking in the UK: An Empirical Study of Practitioners and Academics', *Benchmarking: An International Journal,* **7**(2): 98-117.

Loos, P. 2000. *'Advanced Information Technology Application in ERP Systems'*, Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 635-639.

Lycett, M. 1999. *'The development of Component-Based Evolutionary Information Systems', PhD Thesis, Department of Information Systems and Computing,* Brunel University, Uxbridge, Middlesex, UK.

Maciaszek, L. 2001. *'Requirements Analysis and System Design',* Addison Wesley, New York, USA.

Makey, P. 1998. *'Enterprise Resource Planning',* Butler Group Limited, Hull, UK.

Mandelbaum, M. and Buzacott, J. 1990. 'Flexibility and Decision Making', *European Journal of Operational Research,* **44:**17-27.

Mansell, G. 1991. 'Action Research in Information Systems Development', *Journal of Information Systems,* **1:**29-40.

Manwani, S. 1999. *'Evaluating Multi-Company ERP Goals'*, Proceedings of 9th Annual BIT Conference, Manchester, UK.

Markus, L. 2000. 'Paradigm Shifts - e-Business and Business/Systems Integration', *Communications of the Association for Information Systems,* **4**(10): 1-44.

Markus, L. and Tanis, C. 1999. 'The Enterprise Systems Experience - From Adoption to Success' *In Framing the Domain of IT Management: Glimpsing the Future Through the Past* (Ed, Zmud, R.) Pinnaflex Educational Resources, Inc, Oklahoma, USA, pp. 173-207.

Marshall, C. and Rossman, G. 1999. *'Designing Qualitative Research',* SAGE Publications, London, UK.

Martin, M. 1998. *'An ERP Strategy', Fortune,* 2 February 1998, pp. 95-97

McFarlan, F. W. 1984. 'Information Technology Changes the Way you Compete', *Harvard Business Review,* **62**(3): 98-103.

Mckenney, J. L., Copeland, D. and Mason, R. 1995. *'Waves of Change: Business Evolution Through Information Systems',* Harvard Business School Press, Boston, Massachusetts, USA.

Meier, M., Fullborn, A. and Mertens, P. 2000. *'Vertical Integration of Business News from the Internet within the Scope of SAP Strategic Enterprise Management (SAP SEM)',* Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 663-668.

Mende, U. 2000. *'Software Development for SAP R/3: Data Dictionary, Abap/4, Interfaces',* Springer Verlag, Berlin, Germany.

Miles, M. B. and Huberman, A. M. 1994. *'Qualitative Data Analysis: An Expanded Sourcebook',* SAGE Publications, Newbury Park, California, USA.

Monson-Haefel, R. 2000. *'Enterprise Javabeans',* O'Reilly & Associates, London, UK.

Montealegre, R. 1998. 'Waves of Change in Adopting the Internet: Lessons from Four Latin America Countries', *Information Technology and People,* **11**(3): 235-260.

Mooney, J. 1995. *'Portability and Reusability: Common Issues and Differences',* Proceedings of Proceedings of the ACM 23rd annual conference on computer science, pp. 150-156.

Morgenthal, J. 1998. 'The Distributed Java Platform', *Information Systems Management,* **15**(1): 16-20.

Morgenthal, J. 1999. *'Transforming Business Data into Business Events', EAI Journal,* www.eaijournal.com

Morgenthal, J. and La Forge, B. 2000. *'Enterprise Application Integration with XML and Java',* Prentise-Hall, Inc., New Jersey, USA.

Motsios, T. 1999. *'Implementing ERP systems', MSc,* Athens University of Economics and Business, Athens.

Mowbray, T. J. and Zahavi, R. 1995. *'The Essential CORBA',* John, Wiley and Sons Inc., New York.

Muhanna, W. 1994. 'SYMMS: A Model Management System that Supports Model Reuse, Sharing and Integration', *European Journal of Operational Research,* **72**(2): 214-243.

Nath, P. and Mrinalini, N. 2000. 'Benchmarking the Best Practices of Non-Corporate R&D Organisations', *Benchmarking: An International Journal,* 7(2): 86-97.

Natis, Y., Pezzini, M. and Schulte, R. 1999. *'Middleware Deployment Trends: Survey of Real-World Enterprise Applications',* Gartner Group, New York, USA.

Niederman, F., Brancheau, J. C. and Wetherbe, J. C. 1991. 'Information System Issues for the 1990s', *MIS Quarterly,* **15**(4): 475-500.

Nissen, M. 2000. *'Supply Chain Process and Agent Design for E-Commerce',* Proceedings of 33rd Hawaii International Conference on System Sciences, Hawaii, USA, [CD Proceedings].

Noffsinger, W., Niedbalski, R. and Blanks, M. 1998. 'Legacy Object Modelling, Speeds Software Integration', *Communications of the ACM,* **41**(12): 80-89.

Nolan, R. L. 1973. 'Managing the Computer Resources: A Stage Hypothesis', *Communications of the ACM,* **16**(7): 339-405.

Nolan, R. L. 1979. 'Managing the Crises in Data Processing', *Harvard Business Review,* **5**:115-126.

O'Callaghan, A. 1999. 'Migrating Large Scale Legacy Systems to Component-Based and Object-Technology', *Communications of the Association for Information Systems,* **2**(article 3): 1-39.

O'Leary, D. 2000. *'Enterprise Resource Planning Systems',* Cambridge University Press, Cambridge, UK.

Oliver, D. and Romm, C. 2000. *'ERP Systems: The Route to Adoption',* Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 1039-1044.

Orlikowski, W. and Baroudi, J. 1991. 'Studying Information Technology in Organisations: Research Approaches and Assumptions', *Information Systems Research,* **2**(1): 1-28.

Pant, S. and Ravichandran, T. 2001. 'A Framework for Information Systems Planning for e-business', *Logistics Information Management,* **14**(1/2): 85-98.

Parr, A. and Shanks, G. 2000. *'A Taxonomy of ERP Implementation Approaches',* Proceedings of Thirty-Third Annual Hawai International Conference on System Sciences, HICSS 33, Maui, Hawaii, [CD Proceedings].

Pawar, K. and Driva, H. 2000. 'Electronic Trading in a Supply Chain: A Holistic Implementation', *Logistics Information Management,* **13**(1): 21-32.

Pender, L. 2000. *'EAI vendors bridge best-of-breed gap', PC Week,* 24th January 2000, pp. 32-33

Phillips, E. M. and Pugh, D. S. 1994. *'How to Get a PhD: A Handbook for Students and Their Supervisors',* Open University, London, UK.

Porter, M. E. and Miller, V. E. 1985. 'How Information Gives You Competitive Advantage', *Harvard Business Review,* **63**:149-160.

Puschmann, T. and Alt, R. 2001. *'Enterprise Application Integration - The Case of the Robert Bosch Group',* Proceedings of the 34th Hawaii International Conference on System Sciences, Maui, Hawaii, USA, [CD Proceedings].

Rackoff, N., Wiseman, C. and Ullrich, W. A. 1985. 'Information Systems for Competitive Advantage: Implementation of a Planning Process', *MIS Quarterly,* **9**.

Radding, A. 1999. *'XML: The Language of Integration', Information Week Online,* http:www.informationweek.com/759/xml.htm

Ramanath, A. M. 2000. *'The Role of Information Systems Development Methods in Interorganisation Systems Development', Department of Information Systems and Computing,* Brunel University, London, UK.

Rao, S. S. 2000. 'Enterprise Resource Planning: Business Needs and Technologies', *Industrial Management & Data Systems,* **100**(2): 81-88.

Remenyi, D. 1991. *'The Formulation and Implementation of Strategic Information Systems',* Henley Management College, Brunel University, London, UK.

Riggins, F. and Rhee, H. 1998. 'Toward a Unified View of Electronic Commerce', *Communications of the ACM,* **41**(10): 88-95.

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

Robertson, P. 1997. 'Integrating Legacy Systems with Modern Corporate Applications', *Communications of the ACM,* **40**(5): 39-46.

Robertson, S. and Robertson, R. 1999. *'Mastering the Requirements Process',* Addison Wesley, New York, USA.

Roethlisbeger, F. J. 1977. *'The Elusive Phenomena',* Harvard Business Review, Boston, Massachusetts, USA.

Rogers, E. M. 1995. *'Diffusion of Innovations',* Free Press, New York, USA.

Roman, E., Ambler, S. and Jewell, T. 2001. *'Mastering Enterprise JavaBeans',* John Wiley & Sons Inc., New York, USA.

Rosen, M. 1998. 'Understanding COM/CORBA Interworking', *ACM StandardView,* **6**(1): 44-49.

Ross, J. 1998. *'The ERP Revolution: Surviving Versus Thriving',* Centre for IS Research, Sloan School of Management, MIT, Boston, Massachusetts, USA.

Rowley, D. 1996. 'The Business of Application Portability', *ACM StandardView,* **4**(2): 80-87.

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

Sanday, P. 1996. 'The Ethnographic Paradigm(s)', *Administrative Science Quarterly,* **24**:527-538.

Sanders, R. 1998. *'ODBC 3.5 Developer's Guide',* McGraw-Hill Professional Publishing,

Santarek, K. 1998. 'Organisational Problems and Issues of CIM Systems Design', *Journal of Materials Processing Technology,* **76**(1-3): 219-226.

Schonefeld, M. and Vering, O. 2000. *'Enhancing ERP- Efficiency through Workflow-Services'*, Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 640-645.

Seeley, R. 1999. *'20 questions for Gartner Group's Roy Schulte', EAI Journal,* www.eaijournal.com

Selsikas, P. 1999. 'Information Management in Process-Based Organisations: A Case Study at Xerox Ltd', *Information Systems Journal,* **9**:181-195.

Serain, D. 1999. *'Middleware',* Springer, Berlin, Germany.

Shang, S. and Seddon, P. 2000. *'A Comprehensive Framework for Classifying the Benefits of ERP Systems'*, Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 1005-1114.

Sharma, R., Stearns, B. and Ng, T. 2001. *'J2EE Connector Architecture and Enterprise Application Integration',* Addison Wesley, Massachusetts, USA.

Sherlund, R., Chiang, S., Meisner, A. and Ma, Y. 1999. *'Technology Enterprise Software',* Goldman Sachs Investment Research, New York, USA.

Sieber, M. 1999. *'A Recurring Improvisational Methodology for Change Management in ERP Implementation'*, Proceedings of 5th Conference on Information Systems, AMCIS 1999, Milwaukee, Wisconsin, USA.

Signore, R., Creamer, J. and Stegman, M. 1995. *'The ODBC Solution: Open Database Connectivity in Distributed Environments',* Prentice Hall, Englewood Cliffs, New Jersey, USA.

Smethurst, J. and Kawalek, P. 1999. *'Structured Methodology Usage in ERP Implementation Projects: An Empirical Study'*, Proceedings of 5th Conference on Information Systems, AMCIS 1999, Milwaukee, Wisconsin, USA, pp. 219-221.

Smith, H. and Poutler, K. 1999. 'Share the Ontology in XML-Based Trading Architectures', *Communications of the ACM,* **42**(3): 110-112.

Smith, N. 1990. 'The Case Study: A Useful Research Method for Information Management', *Journal of Information Technology,* **5**:123-133.

Smithson, S. and Cornford, T. 1996. *'Project Research in Information Systems: A Student's Guide',* Antony Rowe Ltd., Wiltshire, UK.

Somers, T., Nelson, K. and Ragowsky, A. 2000. *'Enterprise Resource Planning (ERP) for the Next Millennium: Development of an Integrative Framework and Implications for Research'*, Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 998-1003.

Sparling, M. 2000. 'Lessons Learned through Six Years of Component-Based-Development', *Communications of the ACM,* **43**(10): 47-53.

Sprott, D. 2000. 'Componentizing the Enterprise Application Packages', *Communications of the ACM,* **43**(4): 63-69.

Stake, R. 2000. 'Case Studies' In Handbook of qualitative research' (Ed, Lincoln, Y. S.) SAGE Publications, Thousand Oaks, California, USA, pp. 435-454.

Stefanou, C. 2000. *'The Selection Process of Enterprise Resource Planning (ERP) Systems'*, Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 988-991.

Stone, E. 1978. *'Research Methods in Organisational Behaviour,'* Goodyear Publishing, Santa Monica, California, USA.

Stonebraker, M. 1999. *'Integrating Islands of Information', EAI Journal*, September/October 1999, pp. 1-5 www.eaijournal.com

Strauss, A. and Corbin, J. 1990. *'Basics of Qualitative Research, Grounded Theory Procedures and Techniques',* SAGE Publications, California, USA.

Strauss, A. and Corbin, J. 1994. 'Grounded Theory Methodology - An Overview' *In Handbook of Qualitative Research* (Eds, Denzin, N. and Lincoln, Y.) SAGE Publications, London, UK, pp. 273-285.

Strauss, A. and Corbin, J. 1998. 'Grounded Theory Methodology - An Overview' *In Strategies of Qualitative Inquiry* (Eds, Denzin, N. and Lincoln, Y.) SAGE Publications, London, UK, pp. 158-183.

Sumner, M. 1999. *'Critical Success Factors in Enterprise Wide Information Management Systems Projects'*, Proceedings of SIGCPR '99, New Orleans, LA, USA, pp. 297-303.

Sumner, M. and Holstetler, D. 1999. *'Factors Influencing the Adoption of Technology in Teaching'*, Proceedings of 5th Conference on Information Systems, AMCIS 1999, Milwaukee, Wisconsin, USA, pp. 951-953.

Swenson, D. W. and Cassidy, J. 1993. 'The Effect of JIT on Management Accounting', *Cost Management,* Spring.

Tabor, S. 2000. *'Electronic Commerce Adoption and Success: A Study of Organisational Factors and Influences'*, Proceedings of Sixth Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 669-671.

Themistocleous, M. and Irani, Z. 2000. *'Taxonomy and Factors for Information System Application Integration'*, Proceedings of Sixth Americas Conference on Information Systems, AMCIS 2000, Long Beach, California, USA, pp. 955-959.

Themistocleous, M. and Irani, Z. 2001a. 'Benchmarking the Benefits and Barriers of Application Integration', *Benchmarking: An International Journal,* **8**(4): 317-331.

Themistocleous, M. and Irani, Z. 2001b. *'Evaluating Application Integration: An Exploratory Case Study'*, Proceedings of Seventh Americas Conference on Information Systems, AMCIS 2001, Boston, Massachusetts, USA, pp. 1376-1380.

Themistocleous, M. and Irani, Z. 2002a. *'Evaluating and Adopting Application Integration: The Case of a Multinational Petroleum Company'*, Proceedings of Thirty-Fifth Annual Hawaii International Conference on System Sciences, Hicss 35, Big Island, Hawaii, USA, [CD Proceedings].

Themistocleous, M. and Irani, Z. 2002b. 'Evaluating Enterprise Application Integration Technologies: A Novel Frame of References', *European Journal of Operational Research,* In Press.

Themistocleous, M. and Irani, Z. 2002d. *'A Model for Enterprise Application Integration Adoption and Evaluation'*, Proceedings of Second International Conference on Systems Thinking in Management (ICSTM-2002), University of Salford, Manchester, UK, Forthcoming.

Themistocleous, M. and Irani, Z. 2002f. 'Novel Taxonomy for Application Integration', *Benchmarking: An International Journal,* **9**(2): In press.

Themistocleous, M., Irani, Z. and Love, P. 2002. *'Enterprise Application Integration: An Emerging Technology for Integrating ERP and Supply Chains'*, Proceedings of Tenth European Conference on Information Systems (ECIS 2002), Gdansk, Poland, Forthcoming.

Themistocleous, M., Irani, Z. and O'Keefe, R. 2001a. 'ERP and Application Integration: Exploratory Survey', *Business Process Management Journal,* **7**(3): 195-204.

Themistocleous, M., Irani, Z., O'Keefe, R. and Paul, R. 2001b. *'ERP Problems and Application Integration Issues: An Empirical Survey'*, Proceedings of Thirty-Four Annual Hawaii International Conference on System Sciences, Hicss 34, Maui, Hawaii, USA, [CD Proceedings].

Themistocleous, M., Irani, Z., Psannis, K. and Vrehopoulos, A. 2001c. *'Application Integration of Information Technology: Classification of Benefits and Barriers'*, Proceedings of International Workshop on New Models of Business: Managerial Aspects and Enabling Technology, St. Petersburg, Russia, pp. 153-161.

Themistocleous, M., Irani, Z. and Sharif, A. 2000. *'Evaluating Application Integration'*, Proceedings of 7th European Conference on Evaluation of Information Technology (ECITE 2000), Dublin, Ireland, pp. 193-202.

Themistocleous, M. and Poulymenakou, A. 1998. *'Developing Electronic Stores: Some Critical Considerations for practice'*, Proceedings of 9th International workshop on Database and Expert Systems Application (DEXA 98), Vienna, Austria, pp. 673-679.

Timmers, P. 1998. 'Business Models for Electronic Markets', *Electronic Markets,* **8**(2): 3-7.

Timmers, P. 1999. *'Electronic Commerce: Strategies and Models for Business to Business Trading',* John Wiley and Sons, London, UK.

Treese, W. 1998. 'Putting it Together: What's all the noise about XML?' *ACM NetWorker,* **2**(3): 27-29.

Urlocker, Z. 2000. *'Return to eBusiness Integration', EAI Journal,* January 2000, www.eaijournal.com

Usdin, T. and Graham, T. 1998. 'XML: Not a Silver Bullet, but a Great Pipe Wrench', *ACM StandardView,* **6**(3): 125-132.

Van Den Heuvel, W., Papazoglou, M. and Jeusfeld, M. 1999. *'Configuring Business Objects from Legacy Systems'*, Proceedings of 11th International Conference on Advanced Information Systems Engineering (CAISE'99), Heidelberg, Germany.

Van Heck, E. and Ribbers, P. 1999. *'The Adoption and Impact of EDI in Dutch SME's'*, Proceedings of 32nd Hawaii International Conference on System Sciences, Hawaii, USA, [CD Proceedings].

Vankatraman, N. and Henderson, J. C. 1998. 'Real Strategies for Virtual Organisations', *Sloan Management Review,* **Fall**:33-47.

Volkoff, O. 1999. *'Using the Structural Model of Technology to Analyse an ERP Implementation'*, Proceedings of 5th Conference on Information Systems, AMCIS 1999, Milwaukee, Wisconsin, USA, 235-237.

Walsham, G. 1993. *'Interpreting Information Systems in Organisations',* Wiley, Chichester, UK.

Walsham, G. 1995a. 'The Emergence of Interpretivism in IS Research', *Information Systems Research,* **6**(4): 376-394.

Walsham, G. 1995b. 'Interpretive Case Studies in IS Research: Nature and Method', *European Journal of Information Systems,* **4**:74-81.

Ward, J. and Griffiths, P. 1997. *'Strategic Planning for Information Systems',* John Wiley and Sons, Chichester, UK.

Weill, P. and Broadbent, M. 1998. *'Leveraging the Infrastructure',* Harvard Business Scholl Press, Boston, USA.

White, S., Fisher, M., Cattell, R., Hamilton, G. and Hapner, M. 1999. *'JDBC(TM) API Tutorial and Reference: Universal Data Access for the Java(TM) 2 Platform',* Addison-Wesley, Massachusetts, USA.

Wijegunarate, I. and Fernandez, G. 1998. *'Distributed Applications Engineering',* Springer, London, UK.

Wutka, M. 2001. *'Special Edition Using Java 2 Enterprise Edition (J2EE): With JSP, Servlets, EJB 2.0, JNDI, JMS, JDBC, CORBA, XML and RMI',* QUE Publishing,

Yang, J. and Papazoglou, M. 2000. 'Interoperation Support for Electronic Business', *Communications of the ACM,* **43**(6): 39-47.

Yin, R. K. 1994. *'Case Study Research: Design and Methods',* SAGE Publications, California, USA.

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

Zinner, H. 1999. *'Diffusion of Information Systems: Barriers and Drivers for Magnifying EDI in Danish Enterprises', PhD Thesis, Centre for Electronic Commerce,* Copenhagen Business School, Copenhagen, Demark.

Abbreviations

AI	Application Integration
API	Application Programming Interface
APO	Advanced Planner Optimiser
CBA	Cost Benefit Analysis
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
DOT	Distributed Object Technologies
EAI	Enterprise Application Integration
EBI	Extended Enterprise Integration
ebusiness	Electronic Business
e-business I	e-business Integration
e-commerce	Electronic Commerce
eCRM	Electronic Customer Relationship Management
EDI	Electronic Data Interchange
EJB	Enterprise Java Beans
eProcurement	Electronic Procurement
ERP	Enterprise Resource Planning
eSCM	Electronic Supply Chain Management
eStore	Electronic Store
IS	Information Systems
IT	Information Technology
JDBC	Java Data Base Connectivity
MOM	Message Oriented Middleware
ODBC	Open Data Base Connectivity
OO	Object Oriented
ORB	Object Request Broker
ROI	Return On Investment
RPC	Remote Procedure Call
SCI	Supply Chain Integration

SI	Systems Integration
SME	Small Medium Enterprise
TP Monitor	Transaction Process Monitor
VCI	Value Chain Integration
XML	Extensible Markup Language

APPENDIX A: Enterprise Application Integration Terminology

Summary

Literature review as well as integration market analysis indicates that there is a terminology confusion surrounding Enterprise Application Integration area. Terms such as Application Integration (AI) and Enterprise Application Integration (EAI) are used to describe the whole area and are often used interchangeably in a similar fashion to Information Technology (IT) and Information Systems (IS). However, a number of diverse *terms, types, layers, levels, scenarios, mechanisms, architectures, approaches* etc. are also used to describe Enterprise Application Integration. The purpose of this appendix is to summarise and clarify much of the confusing terminology surrounding application integration area.

A1. Definitions

Bibliographical references on integration area indicate that there are various terms and definitions regarding application integration. Definitions are classified into the following categories: (a) Enterprise Application Integration (EAI); (b) Application Integration (AI); (c) System Integration (SI); (d) Supply Chain Integration (SCI); (e) e-business integration and, (f) Extended Business Integration (EBI). The following sections present these definitions.

A1.1. Enterprise Application Integration (EAI)

The majority of research references on application integration approach the whole area using the term *Enterprise Application Integration.* Klasell and Dudgeon (1998) describe that EAI is a new class of system integration that involves the development of new strategic business solutions that integrates functionality from disparate application. According to Linthicum (1999), enterprise application integration:

> *"is the unrestricted sharing of information between two or more enterprise applications. A set of technologies that allow the movement and exchange of information between different applications and business processes within and between organisations."*
>
> Linthicum (1999, p.354)

Ring and Ward-Dutton (1999) note that Enterprise application integration

> *"combines the technologies and processes that enable custom-built and/or packaged business applications to exchange business-level information in formats and contexts that each understand."*
>
> Ring and Ward-Dutton (1999 p.20)

Zahavi (1999) reports that,

> *"EAI targets the integration of varied types of applications that exist in the organisation and between organisations. These include core legacy systems, enterprise resource planning systems (ERP) and newer Web-based applications."*
>
> Zahavi (1999, p. xxxii)

In contrast, literature review has shown that a minority of definitions on EAI argues that enterprise application integration is a technology that is used for package-to-package application integration (Duke *et al.*, 1999). Grimson et *al.* (2000) and Hasselbring (2000) agree with (Duke *et al.*, 1999) that EAI is a package-to-package solution, and it is used to integrate independent Enterprise Resource Planning (ERP) systems. Table A1.1 synopsises definitions on enterprise application integration.

Definition/Description	Reference
"Enterprise Application Integration is the process of placing hardware, software and the business process in the context so that when they are combined the interfaces between components become seamless, information can be easily shared, and systems working together can achieve synergy."	Brown (2000, p.24)
"EAI provides adaptors for leading ERP packages and it is used for package-to-package integration."	Duke *et. al. (1999,* p. 33)
EAI *"enables the real-time movement and exchange of information between different applications within and between organisations in accordance with flexible business rules."*	Edwards and Newing (1999, p.12)
"EAI offers generic adapters for ERP systems and aims at integrating ERP systems within and across enterprises".	Grimson *et. al.* (2000, p.50)
EAI is about integrating *"independent Enterprise Resource Planning systems at this layer"*	Hasselbring (2000, p.34)
"EAI is a new class of system integration that involves the development of new strategic business solutions. These securely integrate functionality from disparate applications."	Klasell and Dudgeon (1998, p.1)
Enterprise application integration *"is the unrestricted sharing of information between two or more enterprise applications. A set of technologies that allow the movement and exchange of information between different applications and business processes within and between organisations"*	Linthicum (1999, p.354)
EAI is: ▪ *"the seamless integration of business processes for the purpose of conducting business electronically.* ▪ *the sharing and/or exchange of data between systems for the purpose of providing a unified interface."*	Morgenthal and La Forge (2000, p.16)
"Enterprise application integration (EAI) combines the technologies and processes that enable custom-built and/or packaged business applications to exchange business-level information in formats and contexts that each understand."	Ring and Ward-Dutton (1999, p.20)
"EAI is the creation of new strategic business solutions by combining the functionality of an enterprise's existing applications, commercial packaged applications, and new code using common middleware."	Ruh *et al. (2000, p. 2)*
"EAI targets the integration of varied types of applications that exist in the organisation and between organisations. These include core legacy systems, enterprise resource planning systems (ERP) and newer Web-based applications."	Zahavi (1999, p. xxxii)

Table A1.1: Summary of Definitions on Enterprise Application Integration

A1.2 Application Integration (AI)

According to Sprott (2000) integration market space grew quickly to form what was initially referred to as enterprise application integration, and more recently simply *Application Integration*. Sprott (2000) defines Application Integration as

> *"the ability to integrate applications with other packaged, built, and legacy applications now and in future."*
>
> Sprott (2000, p.68).

Duke *et al.*(1999) propose a more technical definition and define application integration as

> *"The transportation and transformation of information between one or more applications. The timing and sequencing rules that govern when the transportation and transformation takes place. The integrity constraints that determine the success or failure of the integration."*
>
> Duke *et. al.* (1999, p.17).

A summary of definitions on Application Integration is presented in table A1.2

Definition	Reference
Application Integration "is *the requirement to integrate into new business processes the functional behaviour or business rules of disparate systems or components of them as well as, but not just, the data that underlies them.*"	Duke *et. al.* (1999, p.17)
Application Integration is *"the ability to integrate applications with other packaged, built, and legacy applications now and in future."*	Sprott (2000, p.68)

Table A1.2: Summary of Definitions on Application Integration

A1.3 Systems Integration (SI)

Grimson *et al.* (2000) and Hasselbring (2000) adopt the term Systems Integration (SI) to refer to the same integration area as Duke *et al.* (1999) (Application Integration). Hasselbring (2000) notes that *Systems Integration* *"aims at building applications that are adaptable to business and technology changes while retaining legacy applications and legacy technology as reasonable as possible."*

Hasselbring (2000, p.36).

According to Markus (2000)

"Systems Integration refers to the creation of tighter linkages between different computer-based information systems and databases. Systems Integration is often required to achieve business integration."

Markus (2000, p.10).

Business integration is about integrating business process with information infrastructure (Brown, 2000). Definitions on systems integration are summarised in table A1.3.

Definition	Reference
"System Integration (SI) is about integration various types of systems and applications."	Grimson *et al.* (2000, p.49)
System Integration (SI) *"aims at building applications that are adaptable to business and technology changes while retaining legacy applications and legacy technology as reasonable as possible."*	Hasselbring (2000, p.36)
"Systems Integration refers to the creation of tighter linkages between different computer-based information systems and databases. Systems Integration is often required to achieve business integration."	Markus (2000, p.10)

Table A1.3: Summary of Definitions on Systems Integration

A.1.4 Value Chain Integration (VCI) or Supply Chain Integration

Linthicum (1999) expands the definitions described in previous sections and deal with : (a) enterprise application integration; (b) application integration and, (c) systems integration. Linthicum (1999) proposes a new category of integration called *Supply Chain* or *Value Chain Integration (VCI). Value Chain Integration* incorporates applications of the same value chain across companies. In particular

> *"Supply Chain Integration is the process of joining systems that may exist in two or more enterprises."*
>
> Linthicum (1999, p.267).

According to Yang and Papazoglou (2000)

> *"Value Chain Integration means that an enterprise's business system can no longer be confined to internal processes, programs, and data repositories, rather they must incorporate with other such systems that support links in the supply chain."*
>
> Yang and Papazoglou (2000, p. 47).

Definitions on Value Chain Integration are synopsised in table A1.4.

Definition	Reference
Value or Supply *"Chain Integration is the process of joining systems that may exist in two or more enterprises."*	Linthicum (1999, p.267)
"Value Chain Integration means that an enterprise's business system can no longer be confined to internal processes, programs, and data repositories, rather they must incorporate with other such systems that support links in the supply chain."	Yang and Papazoglou (2000, p. 47)

Table A1.4: Summary of Definitions on Value Chain Integration

A2. Evaluation of Integration Terminology

The definitions reported in the previous sections are classified by the author into two main categories of definitions namely: (a) Enterprise Application Integration and, (b) Application Integration. The contradiction that the various definitions present have led the author to summarise these two categories using two different approaches (perspectives).

The first approach suggests that there are two different meanings around *Application integration.* Application integration (or SI) is a meaning that describes the integration problem. A set of software solutions such as application servers, workflow, screen wrapping tools etc can be used to solve part of this problem. The second meaning is enterprise application integration, which addresses *only* a specific application integration problem; *the problem of package-to-package application integration.*

The second approach suggests that enterprise application integration defines the integration problem that is faced by companies when they attempted to incorporate enterprise or cross-enterprise applications. A new generation of software (EAI software) is used to address this problem. EAI software consists of EAI tools and solutions (e.g. EAI adapters). EAI often includes a number of existing technologies such as middleware, workflow etc. EAI solutions not only cover the package-to-package application integration problem but also many other dimensions of EAI problem. Sprott (2000) clarifies this terminology confusion by reporting that application integration is a term that is used today to describe an area that was initially referred as EAI. Edwards and Newing (2000) prove that EAI does not *only* provide solution to package-to-package application integration. To prove this statement, they present various case studies in which EAI software was used to integrate enterprise and cross-enterprise applications. Table A1.5 synopsises all these case studies based on the integration problem they focus.

Integration Problem	Company	
Autonomous applications integration	▪ EDS Enterprise Solutions ▪ US West ▪ BT Cellnet	▪ Fusitsu Computers ▪ Tesco
Business Processes (or workflow) Integration	▪ Deutsche Bank	
Component Integration	▪ Deutsche Bank	
Customer Relationship Management Integration	▪ US West	
Data Integration (Database Integration, Data Consistency)	▪ British Airways ▪ Honeywell Europe ▪ Elsevier Science	▪ Fusitsu Computers ▪ Tesco
e-business integration	▪ General Motors	
ERP Integration	▪ EDS Enterprise Solutions	▪ VF Corporation
Product oriented applications	▪ Zurich Financial Services	
Supply Chain Integration	▪ Dell	▪ Cisco

Table A1.5: Integration Problems Addressed by EAI Case Studies

Enterprise application integration as a term has a *limited* meaning as it focuses on the integration of intra-organisational applications. However, inter-organisational systems such as supply chain management exist and are shared by two or more enterprises (Kalakota, 2000). These applications do not serve or belong to one enterprise but to a set of companies-partners. The term enterprise application integration cannot be used to describe these categories of shared systems. Consequently, the term Application Integration (AI) has a broader meaning and it is suggested to describe the whole application integration area. However, the use of the abbreviation AI contradicts with the abbreviation that is used to refer to Artificial Intelligence. Thus, in the context of this dissertation the terms application integration and enterprise application integration will have the same meaning with the author using the abbreviation EAI to refer to the technology that integrates enterprise and cross enterprise applications.

A.2.1 Integration Approaches

Confusing terminology surrounding application area as well as the lack of common architectures and approaches has led analysts to propose various architectures to describe and better understand this area (Zahavi, 1999). A plethora of integration approaches, layers, levels, architectures, scenarios, models etc. exists in normative literature. This section focuses on the analysis of the main integration approaches.

Duke *et al.* (1999) report that application integration requires communication at: (a) Business Architecture Layer; (b)Application Architecture Layer and, (c) Technology Architecture Layer to achieve integration since, there is a lack of standards and common architectures. Brown (2000) supports Duke *et al.* (1999) approach but divides the Business Architecture layer into Business Processes Layer and Information Architecture Layer. Along similar lines Hasselbring (2000) refers to the same layers as Duke *et al.*(1999) using the term *Vertical Fragmentation of Organisational Units* and reports that:

- Business Architecture Layer *defines the organisational structure and the workflow for business processes and rules.*

- Application Architecture Layer *is the glue between Business Architecture Layer and Technology Architecture Layer. This layer defines the actual implementation of business concepts.*
- Technology Architecture Layer *defines the information and the communication infrastructure*

Moreover, Hasselbring (2000) notes that this categorisation does not adequately reflects the reality as there is highly interrelation between the business processes of co-operating units and therefore it is important to consider all architecture layers when integrating systems. In addition, Hasselbring (2000) expands this architecture by adding one horizontal layer for each vertical. This expanded architecture is illustrated in Figure A1.1 and includes three layers: (a) inter-organisational processes, (b) enterprise application integration and, (c) middleware layer.

*Inter-organisational Processes Laye*r cuts business processes horizontally through the traditional organisation structure and seeks to organise them in a competitive way. The integration of information systems within the organisations supports inter-organisational business processes as it makes the integration of inter-organisational processes easier.

Enterprise application integration Layer seeks to integrate independent Enterprise Resource Planning systems. This is achieved through messaging services. Applications need to understand the data exchanged.

Middleware Integration Layer uses techniques that integrate componentized information systems with state of the art technologies such as DCOM, CORBA, application servers and database gateways. Middleware integration exchanges data between applications while EAI layer translates the data. However, it is difficult to distinguish the borderline between middleware integration and EAI.

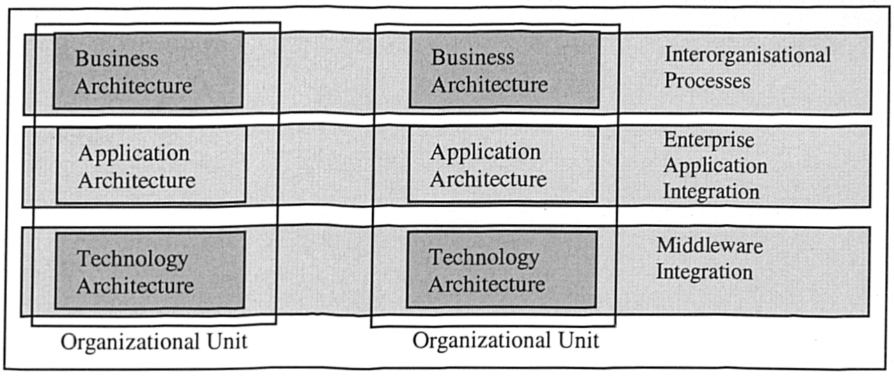

Figure A1.1. Horizontal Integration (Source: Hasselbring, 2000)

Grimson et. al. *(2000), refer approximately to the same categorisation as Hasselbring using the following 4 different approaches:*

- *Distributed Component Based approach* that supports business processes by encapsulating both data and functions using a set of common and domain specific services. This approach requires a high degree of standardisation;
- *Enterprise Application Integration* that is used to integrate specific ERP systems within and across companies;
- *Data Warehousing* that allows data from disparate applications to be integrated and homogenised in data warehouse but it can not support a real time environment and it is a read-only mode; and,
- *Messaging* approach that aims at providing loose coupling of applications. Messaging is easy to be implemented but it is not scaleable and requires a high degree of interface engineering and support.

To support integration requirements such as process integration, data integration, information consolidation and data synchronization, the following mechanisms are available (Duke *et al.*, 1999):

- *Call Interface mechanisms.* Application Programmable Interfaces (APIs) are used to provide interoperability;
- *Messaging mechanisms.* Applications are integrated by sending and receiving messages using queuing mechanisms; and,

- *Data Access and File Transfer mechanisms.* Applications are integrated by direct access to their database or via file transfers.

It is obvious that these mechanisms can facilitate both Hasselbring's (2000) layers and Grimson *et al.* (2000) approaches (e.g Call Interface mechanism supports EAI approach and EAI layer). Similar to these approaches are the Linthicum's (1999) types of integration. Linthicum (1999) describes the following types for EAI:

- *Data-level EAI is used for data integration and it* extracts information from one database. In some cases it processes that information, and updates it in another database. It may also include the transformation and application of business logic to the data that is being extracted and loaded;
- Application-interface level EAI is most applicable to package-to-package application integration and it refers to the leveraging of interfaces exposed by packaged applications (or custom applications);
- *Method-level EAI* is the sharing of the business logic that may exist within the enterprise (e.g. the method for updating a customer record may be accessed from any number of applications, and applications may access each other's methods without having to rewrite each method within the respective application). Method-level EAI can be used for business processes integration; and,
- *User-Interface level EAI* is suitable for custom applications integration. Using this level, developers are able to bundle applications by using their user interfaces as a common point of integration (screen scraping). This level is not covered by the approaches mentioned above.

Ruh *et al.*(2000) use three models similar to Linthicum (1999) levels. The models propose by Ruh *et al.*(2000) deal with: (a) Data Integration Model; (b) Presentation Integration Model and, (c) Functional Integration Model. These models describe the same areas as Linthicum (1999) Data level EAI, User-interface level EAI and method (and Application Interface) integration level respectively. Furthermore, Ring and Ward-Dutton (1999) use similar concepts to (Linthicum, 1999) levels to report four EAI levels, which include: (a) data level, (b) object-level, (c) internal process level and, (d) cross-enterprise process level integration. A basic difference between Ring and Ward-Dutton's approach and the approaches described above is that the former divide business process in cross-enterprise process integration and internal process integration.

Edwards and Newing (2000), Klasell and Dudgeon (1998) and Duke *et al.* (2000) explain application integration area using more technical approaches. Edwards and Newing (2000), focuses on 3 layers:

The business logic layer provides the capability of representing business processes usually using tools that are similar to workflow. Process automation enables the passing of information between systems in accordance with predefined rules to satisfy certain business objectives.

Integration services layer uses message brokering technology to move messages from one type of application to another by changing or translating the format of the message in order to accommodate the needs of the target application. To achieve this functionality EAI vendors minimise the changes required within both source and target *system by developing* series of adapters. Adapters are placed between message broker and the source or target application to hide further the complexity of the interface. Using a library of standard templates, business link more easily their applications without having to implement their interfaces. Many EAI vendors developed series of adapters that offer pre-programmed formats enabling links to be made with standards, databases, legacy systems, networks and business applications. The combination of message brokers *and adapters allows companies* to contemplate technology enabled supply chain integration.

The Messaging Layer is used for message delivery. Many EAI vendors use existing products for message delivery such use of middleware tools. However, many traditional middleware products are not suitable for EAI.

Klasell and Dudgeon (1998) describe the same layers as Edwards and Newing (2000) using the terms: (a) Transport layer, (b) Message Brokering and Translation layer and, (c) Process Automation layer respectively. Likewise, Duke *et. al.* (2000) refer to same concepts using five layers instead of three. This approach is called *Integration Implementation Layer Mode* and includes:

- Transport layer dealing with the physical delivery of information;
- Transaction layer dealing with the integrity and management of transactions that involve integration between applications;
- Transformation layer which, converts (formats) the information between the applications being integrated;
- Timing layer dealing with triggering or activating the exchange of information; and,
- Process layer dealing with the business rules that determines the integration.

All approaches presented in this section are summarised in Table A1.6.

Approach	Applicability	Supported Technologies	References
Integration Processes Layer	Integration of Business Processes	Workflow tools	Hasselbring (2000)
Enterprise Application Integration Layer	Integration of autonomous ERP	EAI solutions, Message services, XML	Hasselbring (2000)
Middleware Integration Layer	Integration of componentized information systems	DCOM, CORBA Transaction Monitors Database Gateways	Hasselbring (2000)
Messaging Integration Approach	Integration of loose coupling of systems	Message services XML	Grimson *et. al.* (2000)
Enterprise Application Integration Approach	Integration of specific ERP systems	Wrapping Techniques XML, Message services	Grimson *et. al.* (2000)
Data Warehouse Integration Approach	Data integration	Data Warehouse	Grimson *et. al.* (2000)
Distributed Component Based Approach	Integration of business processes	DCOM, CORBA	Grimson *et. al.* (2000)
Integration Implementation Model which includes Transport, Transaction, Translation, Transformation, Time and Process layer	Integration of intra and inter-organisational applications	EAI Tools	Duke *et. al,* (1999)
Transport layer	Data integration Application integration	Middleware products EAI solutions	Klasell and Dudgeon (1998)
Translating and Formatting layer	Data integration Application Integration (e.g. ERP-to-ERP)	EAI Adapters, objects, message services Message Brokers, XML	Klasell and Dudgeon (1998)
Process Automation layer	Application integration E-business integration	EAI products	Klasell and Dudgeon (1998)
Data-level EAI	Data integration	XML	Linthicum (1999a)
Application-interface level	Package AI (e.g. ERP-to-ERP)	APIs CORBA, COM	Linthicum (1999a)
Method-level EAI	Business Process integration	EAI adapters, CORBA, COM, MOM, Message brokers	Linthicum (1999a)
User-interface level EAI	Custom Packages (e.g. legacy) Integration	Screen Wrapping	Linthicum (1999a)
Messages and Transportation Layer		Message oriented Middleware Products	Edwards and Newing (2000)
Integration Service Layer		Message Brokers	Edwards and Newing (2000)
Business Logic	Process automation integration	Workflow tools EAI solutions	Edwards and Newing (2000)
Data-level integration	Front-end integration	Middleware products	Ring and Ward-Dutton (1999)
Object level integration	Synchronisation of data between applications and databases	CORBA, COM, Middleware	Ring and Ward-Dutton (1999)
Internal process level integration	Semantic integration	EAI adapters, Middleware	Ring and Ward-Dutton (1999)
Cross-enterprise application integration	E-business integration Package to package integration	EAI adapters, XML	Ring and Ward-Dutton (1999)
Data Integration Model	Database Integration	Database Middleware	Ruh *et al.* (2000)
Presentation Integration Model	Legacy Integration	Screen Wrappers	Ruh *et al.* (2000)
Functional Integration Model	E-business integration Package to package integration Business Processes integration	APIs, EAI adapters	Ruh *et al.* (2000)

Table A1.6: Integration Approaches

The approaches summarise in table A1.6 are classified by the author in two categories. The first classification focuses on the scope of each layer (in terms of system types that are integrated e.g. custom application integration etc). The second one takes into account the technical layers (e.g. transport layer). Both of these classifications are presented in Tables A1.7 and table A1.8 respectively.

Integration Layer	Approach	Reference
Processes Integration	Integration Processes Layer	Hasselbring (2000)
	Process Automation layer	Klasell and Dudgeon (1998)
	Internal process level integration	Ring, and Ward-Dutton, 1999
	Functional Integration Model	Ruh *et al.*(2000)
E-business Integration	Cross-enterprise application integration	Ring, and Ward-Dutton (1999)
Custom applications integration	User-interface level EAI	Linthicum (1999a)
	Presentation Integration Model	Ruh *et al.* (2000)
Package to Package Integration	Enterprise Application Integration Layer	Hasselbring (2000)
	EAI Approach	Grimson *et al.* (2000)
	Application-interface level	Linthicum (1999a)
Components Integration	Distributed Component Based Approach	Grimson *et al.* (2000)
	Middleware Integration Layer	Hasselbring (2000)
Data Integration	Data-level EAI	Linthicum (1999a)
	Data Warehouse Integration Approach	Grimson *et al.* (2000)
	Data-level integration	Ring, and Ward-Dutton (1999)
	Object level integration	Ring, and Ward-Dutton (1999)
	Data Integration Model	Ruh *et al.*(2000)

Table A1.7: Classification of Scope Integration Approaches

Integration Layer	Approach	Reference
Processes Automation	Process Automation layer	Klasell and Dudgeon (1998)
	Business Logic	Edwards and Newing, (2000)
	Time and Process layer	Duke *et al.* (1999)
Translating and Formatting	Translating and Formatting layer	Klasell and Dudgeon (1998)
	Integration Service Layer	Edwards and Newing, (2000)
	Transaction, Translation, Transformation	Duke *et al.* (1999)
Transport	Transport layer	Klasell and Dudgeon (1998)
	Messages and Transportation Layer	Edwards and Newing, (2000)
	Transport	Duke *et al.* (1999)

Table A1.8: Classification of Technical Integration Approaches

APPENDIX B: Integration Technologies

B1. Middleware Vs Enterprise Application Integration

During the last decade, various technologies were introduced to achieve integration between computer platforms, applications and networks. Initially, these technologies formed individual software categories such as protocols and programming interfaces. Later on the majority of integration tools were promoted as part of *Middleware* technology. The last few years, a new generation of integration software, called Enterprise Application Integration (EAI), addresses many incorporation issues and it is promoted as a promising solution to integration problems (Morgenthal and La Forge, 2000; Ruh *et al.*, 2000). EAI technology becomes more popular and EAI market is rapidly expanded (Forrester Research, 1998). As a result, middleware vendors started promoting their products as EAI solutions and caused a terminology and market confusion. The purpose of this section is to clarify the confusion between EAI and middleware technology.

Mowbray and Zahavi (1995) report that the term *middleware* was initially used to describe database products that could communicate with diverse databases. Later on it was used to describe the function of Object Request Brokers (ORB) and Remote Call Procedures (RPC) in the integration process (Zahavi, 1999). Nowadays, the term middleware is used to describe a variety of tools and technologies that sit in a middle layer below applications and above networking and operating system software. Linthicum (1999a) defines middleware as

> *"a mechanism that allows one entity (application or database) to communicate with another entity or entities"*
>
> Linthicum (1999a, p. 119).

Based on this definition, the majority of integration tools (including EAI) can be described as middleware technology. However, literature supports that EAI is not the same as middleware technology and many differences exist between them (and especially between traditional middleware and EAI) (Duke *et al.*, 1999; Klasell and Dudgeon, 1998; Zahavi, 1999). Traditional middleware technology (such as RPCs) cost more and leads to non-flexible solutions. Edwards and Newing (2000) report that EAI allows flexible integration in a way that is different from middleware although EAI is based on many of middleware strengths. Ruh *et al.*(2000) mention that EAI comprises four building blocks including middleware. According to Ring and Ward-Dutton (1999) integration happens at three levels: *Data, Object* and *Process* level integration. Middleware technology provides integration solutions for data level and part of object level where EAI covers all three levels. As mentioned above, EAI is a combination of middleware, workflow and data transformation technologies. Middleware offers connectivity services to an EAI solution, workflow technology provides process management services and, transformation products offer data transformation services (Ring and Ward-Dutton, 1999).

Enterprise application integration is based on a diversity of integration technologies which are classified by Linthicum (1999a; 1999b) into: (a) database oriented middleware; (b) message based; (c) transaction based (d) distributed object and, (e) interface oriented technologies. The next sections present the basic features of Integration technologies.

B2. Database Oriented Middleware

Database Oriented Middleware is the software that connects an application to a database. It is fundamental for application integration as most of applications data are stored in databases that are only accessible using this category of middleware technology (Linthicum, 1999b). Database Oriented Middleware provides the ability to access databases and data files on another computer using a well-defined Application Programming Interface (API) (Ruh *et al.*, 2000). Initially, each database vendor provided its own solution for distributed data access. Nowadays, Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC) have become popular standard mechanisms for access to distributed databases. These mechanisms are described below.

B2.1 Open Database Connectivity (ODBC)

Ruh *et al.*(2000)) define ODBC *"a standard interface originally intended for relational database management systems."* (Ruh *et al.*, 2000), p.54). ODBC provides a well defined and database independent API that simplifies database access from windows as well as from other operating systems. ODBC exposes a single API to facilitate access to a database and then determines the appropriate ODBC driver to support the translation of data from the source database to the target. It supports the translation layer and enables an EAI solution to move quickly data from one database to the other. ODBC focuses more on relation databases than other types of databases such as multidimensional, object-oriented or hierarchical databases. It should be consider when operating in a multi-database environment that requires access to several databases from the same application or integration server (e.g. Message Broker, application server). ODBC is a stable, mature mechanism that allows high performance database access (Brown, 2000).

OPEN DATA BASE CONNECTIVITY (ODBC)	
Advantages	**Reference**
Database independence	Linthicum (1999a)
High performance	Brown (2000)
Supports the translation layer	Signore *et al.*(1995)
Collaborates with DOT, MOM, CORBA, TP monitors, JDBC	Ruh *et al.*(2000)
Stable	Signore *et al.*(1995)
Mature	Serain (1999)
Simplifies database access	Linthicum (1999b)
Disadvantages	**Reference**
Does not supports the transportation layer	Serain (1999)
Does not supports process automation layer	Sanders (1998)

Table B.1 ODBC - Advantages and Disadvantages

B2.2 Java Database Connectivity (JDBC)

JDBC is also a stable approach that provides a standard Java-enabled database API. Its functionality is similar to ODBC and it provides access to most relational databases from Java-enabled applications and environments. It also provides database access for many EAI enabled products such as application servers, message brokers and message oriented middleware (MOM) (Ruh *et al.*, 2000). Although both JDBC and ODBC simplify database access, they do not solve the EAI problem since the data must be distributed, identified, classified, and altered to reach the target application (Linthicum, 1999b). Thus, both JDBC and ODBC (and Database Oriented Middleware in general) have to collaborate with other technologies in order to achieve this functionality. As a result, database oriented middleware need to collaborate with Distributed Object Technology (DOT) (e.g. CORBA, DCOM), MOM, Transaction Process Monitors (TP monitors), message brokers and applications servers to facilitate data transportation and transformation (Linthicum, 1999b).

JAVA DATA BASE CONNECTIVITY (JDBC)	
Advantages	**Reference**
Supports translation layer	Wutka (2001)
Stable	White *et al.*(1999)
Mature	Linthicum (1999a)
Collaborates with DOT, MOM, CORBA, TP monitors	Ruh *et al.*(2000)
Supports Java environments	Morgenthal and La Forge (2000)
Disadvantages	**Reference**
Does not supports transport layer	Linthicum (1999b)
Does not supports process automation layer	Wutka (2001)

Table B.2: JDBC - Advantages and Disadvantages

B3. Messaged Based Technologies

Message oriented software manages the distribution of messages from one application to the other (Edwards and Newing, 2000) and uses synchronous or/and asynchronous types of communications (Bernstein, 1996; Ring and Ward-Dutton, 1999; Zahavi, 1999). The type of communication determines a kind of dependencies among two applications and influences the processing sequence of the involved applications (or components). In *asynchronous* communication, applications communicate over time without having to wait for the receiver to receive and process the message (Morgenthal and La Forge, 2000). This allows the sender application to continue processing after sending its request (message) to the target application (Duke *et al.*, 1999). In *synchronous* messaging systems, there is a high degree of coupling (Edwards and Newing, 2000) as the sender pauses its operations and waits for the receiver to process the message and reply (Ruh *et al.*, 2000). This type of communication is accomplished in a co-ordinate manner.

Apart from the type of communication, three connection types (Point-to-Point, Publish/Subscribe, Hub and Spoke) exist and they are used for the distribution of messages. In *Point-to-Point* connections, two applications are connected directly to one another using a simple communication channel (pipe) (Duke *et al.*, 1999; Linthicum, 1999a; Ruh *et al.*, 2000). In *Publish/Subscribe* connections, a number of applications are connected together (Ring and Ward-Dutton, 1999). Each application defines the information, the data structures and the types of requests it is interested in receiving and, decides about which events it wants to be notified (Ruh *et al.*, 2000). One application (publisher) publishes information on a subscription list, which can be accessed by all applications. One or more applications can express (by subscribing to the list) their indention to receive this information (Duke *et al.*, 1999; Morgenthal and La Forge, 2000). A variation of Publish/Subscribe model is the *Bus* or *Broadcast* connection where all messages are broadcast to the other applications using a central communication channel (bus). In this type of connection there is no central subscription list (hub) (Duke *et al.*, 1999). In *Hub and Spoke* connections, a number of applications can connect to a central hub that contains the rules for connecting application together (Duke *et al.*, 1999). All applications send their messages to the hub, which distributes the right message to the right receiver using its rules (Morgenthal and La Forge, 2000). According to (Morgenthal and La Forge, 2000) hub and spoke communication model can be seen as a variation of Publish/Subscribe.

Point-to-point connection model has a high degree of complexity (Duke *et al.*, 1999) as for a number of x applications a total $x\,(x-1)\,/2$ different connections are needed to interconnect all applications together (Themistocleous *et al.*, 2000). In contrast, Publish/Subscribe, Hub and Spoke and Bus (or Broadcast) models reduce the number of connections (only x connections are needed) and therefore decrease the complexity. Hub and Spoke is the most flexible connection model (Linthicum, 1999a), it provides integration efforts but it has low performance as it has to process each message separately in order to send it to the receiver. Publish and Subscribe is also flexible but requires no integration effort and it might not cope with the diversity of applications and the complexity of (integration) rules (Ruh *et al.*, 2000).

Messaged based integration software includes Remote Procedure Calls (RPCs), Message Oriented Middleware (MOM), Message Brokers and Extensible Markup Language (XML). These categories are described below.

B3.1 Remote Procedure Calls (RPC)

A Remote Procedure Call (RPC) is a mechanism that makes message passing look like a procedure call (Bernstein, 1990) and hides the complexities of operating systems and networks through a function call (Linthicum, 1999b). RPCs are focused on the integration of the procedures of distributed applications across a network (Ruh *et al.*, 2000) and based on a synchronous (Birrell and Nelson, 1984; Linthicum, 1999b) Point-to-Point communication and connection model (Edwards and Newing, 2000). The advantages of RPCs are the simplicity of its mechanism and the ease of programming (Linthicum, 1999b).

REMOTE PROCEDURAL CALL (RPCs)	
Advantages	**Reference**
Hides complexities of operating systems and network	Serain (1999)
Supports the transport layer	Edwards and Newing (2000)
Synchronous communication (Supports Real Time transactions)	Birrell and Nelson (1984)
Simple mechanism	Linthicum (1999b)
Ease of programming	Wijegunarate and Fernandez (1998)
Disadvantages	**Reference**
Require high-speed networks	Edwards and Newing (2000)
Require high processing power	Serain (1999)
Non flexible mechanism	Linthicum (1999a)
High maintenance cost	Zahavi (1999)
Point-to-point connections lead to complex solutions	Ruh *et al* (2000)
Complex to replicate the results	Edwards and Newing (2000)
Invasive method	Zahavi (1999)

Table B.3: RPCs - Advantages and Disadvantages

However, the invasive nature of RPCs is a significant drawback and leads to high maintenance costs and complexity (Linthicum, 1999b). In addition, RPCs require high-speed networks, high processing power, and high level of detail technical input and it is complex to replicate the results (Edwards and Newing, 2000). RPC is the only type of Middleware that declines (Ruh *et al.*, 2000), due to its procedural nature as well as due to the fact that other integration technologies such as MOM and DOT (CORBA, DCOM) are more powerful and provide characteristics of RPCs (e.g. DOT products support synchronous communications) (Linthicum, 1999a; 1999b; Ruh *et al.*, 2000).

B3.2 Message Oriented Middleware (MOM)

"Message Oriented Middleware is a type of middleware that uses messages as the method of integration; it provides the ability to create, manipulate, store and communicate these messages." (Ruh *et al.*, 2000) p. 55). According to Linthicum (1999b) MOM was designed to solve many of the RPCs drawbacks. It provides the ability to integrate applications based on messages technology and uses a point-to-point asynchronous messaging mechanism for the communication of applications (Morgenthal and La Forge, 2000; Ruh *et al.*, 2000). As a result, MOM does not block applications processing and ensures delivery of messages (Linthicum, 1999a). In contrast to RPCs, MOM is more flexible, requires less processing power, and has a higher performance. Furthermore, MOM can be used for exchanging data between Java and non-Java applications (Morgenthal and La Forge, 2000).

However, MOM requires altering the source and target application (Edwards and Newing, 2000), which increases maintenance cost and complexity. In addition, MOM cannot be effective for component integration, as messages are not as easily visible as interfaces. Moreover, component integration requires plug and play properties as well as reusability but messages and MOM technology can not fulfil these requirements (Ruh *et al.*, 2000).

MESSAGE ORIENTED MIDDLEWARE (MOM)	
Advantages	**Reference**
More flexible than RPCs	Natis *et al.*(1999)
Higher performance than RPCs	Morgenthal (1999)
Supports the transportation layer	Ruh *et al.*(2000)
Supports the exchanging of data between Java and non-Java environments	Wijegunarate and Fernandez (1998)
Disadvantages	**Reference**
Invasive method	Edwards and Newing (2000)
Does not support real-time integration	Linthicum (1999a)
Does not support the translation layer	Edwards and Newing (2000)
High maintenance cost	Zahavi (1999)
Supports point-to-point approach which is not flexible and leads to complex solutions	Ring and Ward-Dutton (1999)

Table B.4: MOM - Advantages and Disadvantages

B3.3 Message Broker

Zahavi (1999) reports that existing middleware technologies like MOM, ORB and Object Transaction Monitors (OTM) can not adequately support integration. MOM is an invasive method, based on point-to-point communications and cannot be used for message interpretation or semantics. ORBs are synchronous and offer more abstraction using object-oriented interfaces but again cannot support efficient enterprise integration requirements. Object transaction monitors combine ORB, transaction monitors and messaging capabilities but there is still a need for a set of higher-level services to support integration requirements.

EAI vendors extend and improved existing message oriented middleware to include application integration functionality by developing the *Message Broker* (Ring and Ward-Dutton, 1999). A message broker moves messages from one application to the other by changing or translating the format of message in order to support the needs of the target application (Edwards and Newing, 2000). Message exchange is based on an asynchronous hub and spoke communication model, which enables applications to operate independently (Klasell and Dudgeon, 1998). Message brokers are important part of an EAI infrastructure (Klasell and Dudgeon, 1998) as they allow non-invasive links to be made among different applications which results in a more flexible and easier to maintain integration mechanism (Edwards and Newing, 2000). Moreover, message brokers are scalable.

The purpose of message brokers is to integrate multiple business process as well as applications (e.g. custom applications, centralised, distributed, package applications etc) (Linthicum, 1999b). In doing so, they can use adapters to allow companies to achieve application integration (e.g. supply chain integration) (Edwards and Newing, 2000). Message brokers support transformation of data, data types and messages, message filtering and routing. In addition, they provide rules processing capabilities, hosting business functions, message translation engines and bridges to many different platforms and applications (using pre-built EAI adapters or existing APIs) (Linthicum, 1999a).

Message brokers lack of abstraction and OO capabilities because they are based on messaging. According to Zahavi (1999) a combination of message brokers with component capabilities will offer a more robust EAI infrastructure and better address integration problems.

MESSAGE BROKER	
Advantages	**Reference**
Support transportation layer	Klasell and Dudgeon (1998)
Support translation layer	Edwards and Newing (2000)
Support process automation layer	Linthicum (1999a)
Easier to maintain	Edwards and Newing (2000)
Non Invasive	Klasell and Dudgeon (1998)
Scalable	Linthicum (1999b)
Flexible	Duke *et al.*(1999)
Provide many pre-build adapters to bridge disparate systems	Linthicum (2000)
Support the hub and spoke communication model	Klasell and Dudgeon (1998)
Disadvantages	**Reference**
Lack of abstraction	Zahavi (1999)
Does not support real-time transactions	Klasell and Dudgeon (1998)

Table B.5: Message Brokers - Advantages and Disadvantages

B3.4 Extensible Markup Language (XML)

XML is a meta-language that provides a standard mechanism for data exchange between applications and companies (Cingil *et al.*, 2000; Ring and Ward-Dutton, 1999). It provides specifications for designing text formats in a way that produces files that are easy to generate and read (Edwards and Newing, 2000). Moreover, it provides definitions that describe the data structure and meaning (Ring and Ward-Dutton, 1999) and it is a language for representing hierarchically structured information (Morgenthal and La Forge, 2000). Treese (1998) defines XML as *"a formal way to annotating documents to indicate how they should be interpreted, presented or otherwise processed."* (Treese, 1998), p. 28).

XLM uses content oriented tags, which allows XML data to be self-describing. This enables an application to understand the meaning of the data and therefore enhances the ability of remote systems to translate and operate documents exchanged over Internet (Cingil *et al.*, 2000). It separates the content data from information on how this data should be presented or proceeded. Format and processing information are managed separately (Usdin and Graham, 1998). XML uses Resource Description Framework (RDF) to process *metadata and provides* interoperability to applications that exchange XML messages (Cingil *et al.*, 2000; Linthicum, 1999a). RDF imposes syntax and structural constrains in describing resources in order to avoid any ambiguity in expressing metadata (Cingil *et al.*, 2000).

A part of integration deals with the exchanging of data among two or more applications (Morgenthal and La Forge, 2000). When, this kind of integration is happened, knowledge about involved applications is required. XML simplifies integration by allowing systems to exchange information without having knowledge about the participating applications. Source application just translates its data to XML standard and sends it to the target. The receiver extracts the information from XML message and transformed it into its structure (Linthicum, 1999a).

XML is flexible and offers bindings for object models (e.g. CORBA, COM) and other languages as well as supports the communication between components (Morgenthal and La Forge, 2000). It can be used for sharing data among Java and non-Java environments using relational databases (Zahavi, 1999). Moreover, it can work together with Java to develop dynamic applications by binding Java components based on XML document type. This allows companies to build more flexible applications and EAI solutions (Morgenthal and La Forge, 2000). In general, XML can support EAI in two ways. Firstly it can be used as a description language to specify resource interfaces. All messages sent by resource interfaces to a transformation service can be self-describing using XML. As a result, it simplifies the implementation of message validation service. Secondly, it can be used as a metadata standard for object integration (Ring and Ward-Dutton, 1999).

Likewise Electronic Data Interchange (EDI), XML is used for exchanging messages but it makes applications more flexible and easier to be developed and maintained than EDI approach. As a result, XML can be thought as a better and cheaper solution for supply chain integration and inter-organisational integration (Edwards and Newing, 2000; Linthicum, 1999a).

However, XML is not a panacea for EAI (Smith and Poutler, 1999). It has a variety of features that have to be applied in a specific range of IT infrastructure to provide the advantages presented in this section. Many companies that attempted to apply XML to technological areas that are not appropriate had little success as this action practically dilute its real value and results (Linthicum, 2000).

EXTENSIBLE MARKUP LANGUAGE (XML)	
Advantages	**Reference**
Supports transportation layer	Cingil *et al.*(2000)
Supports translation layer	Ring and Ward-Dutton (1999)
Achieves interoperability to XML compatible applications	Cingil *et al.*(2000)
Simplifies integration	Morgenthal and La Forge (2000)
Flexible mechanism	Linthicum (2000)
Supports communication between components	Radding (1999)
Shares data between Java and non Java environments	Zahavi (1999)
Cheap solution for B2B integration	Edwards and Newing (2000)
Maintainable	Smith and Poutler (1999)
Supports data transformation and integration problems	Duke *et al.*(1999)
Disadvantages	**Reference**
Does not solve process integration	Duke *et al.*(1999)
Provides extensibility which may add complexity	Morgenthal and La Forge (2000)
Invasive mechanism	Morgenthal and La Forge (2000)

Table B.6: XML - Advantages and Disadvantages

XML solves many of the data transformation problems but it does not address issues such as process integration (Duke *et al.*, 1999). Other drawback is the fact that during an XML transaction, transformation of data takes places twice (one from source application data to XML message and another one from XML message to target application data). Moreover, during an XML interchange redundant and unnecessary information (e.g. tags, metadata) is transmitted. XML is extensible in a way that users can create their own tags, but if anyone invents new tags it will not possible to achieve interoperability (Cingil *et al.*, 2000).

B.4 Transaction Based Technologies

B4.1 Transaction Process Monitors (TP monitors)

A transaction is defined by Ruh *et al.*(2000) as *"a single unit of work to support one or more business functions that must be completed in a single action to achieve a business purpose. If not all required business functions can be completed, then the transaction not be completed."* Ruh *et al.*(2000, pp. 108-109).

TP monitors are middleware products that preserve the integrity of a transaction (Linthicum, 1999a). According to Bernstein (1990) a transaction process monitor is a mechanism that co-ordinates *"the flow of transaction requests between terminals or other devices and* *application programs that can process these requests."* Bernstein (1990, p. 77). A request deals with one transaction and asks the system to execute this transaction (Bernstein, 1996). In addition, TP monitors facilitate the communication between two or more applications and provide a location for application logic. Application logic is encapsulated within a transaction. If problems occur during the transaction, then the transaction rolls back. For example, a transaction can be formed to update two accounts to effect a transfer of funds. In case the one account cannot be updated, both of them recover their initial data. TP monitors allow a transaction to be formed by a sender and then ensure that it gets to the right place, at the right time and completed in the right order (Ruh *et al.*, 2000). TP monitors provide tools to ensure the integrity of complex business processes by providing atomicity, consistency, isolation and durability of transaction and support features such as auto restart, error logging and replication, fail-over and rollback to eliminate failure (Linthicum, 1999a; Ruh *et al.*, 2000; Zahavi, 1999).

Nowadays, there is a trend for merging TP monitors and DOT (e.g. COM, CORBA etc) as the latter provide TP monitors services (Linthicum, 1999a). Moreover, MOM technology has also incorporated TP monitors functionality (Ruh *et al.*, 2000). TP monitors integrate services that make them accessible through a simplified API (Bernstein, 1996) and can connect with databases, message queues and other applications. The main advantages of TP monitors include: (a) enhanced portability (Bernstein 96); (b) enhanced scalability; (c) fault tolerance mechanism (able to recover if a transaction fails); (d) they support high transaction processing; (e) offer message priority scheduling; and, (f) support large-scale distributed transaction oriented development (Linthicum, 1999b).

However, TP monitors have invasive nature and they are depended on the middleware they use (Bernstein, 96). Furthermore, TP monitors do not support even-driven information processing as well as they do not support content or message transformation services without a lot of programming. In addition, Linthicum (1999b) argues that message brokers and MOM are much better on sharing of data than TP monitors.

TRANSACTION PROCESS MONITORS (TP monitors)	
Advantages	**Reference**
Facilitate the communication of two and more applications	Linthicum (1999a)
Ensure the integrity of complex business processes	Zahavi (1999)
Portable	Bernstein (1996)
Provide enhanced scalability	Bernstein (1996)
Support fault tolerance mechanisms	Serain (1999)
Support high transaction processing	Linthicum (1999b)
Support large scale distributed transaction oriented development	Wijegunarate and Fernandez (1998)
Offer message priority scheduling	Linthicum (2000)
Supports communication with API and databases	Bernstein (1990)
Disadvantages	**Reference**
Invasive	Bernstein (1996)
Do not support process layer	Ruh *et al.*(2000)
Do not support component integration	Linthicum (1999a)
Support the translation layer in a complex and non flexible way	Wijegunarate and Fernandez (1998)

Table B.7: TP monitors - Advantages and Disadvantages

B4.2 Application Servers

Application servers can be considered as the next generation of TP monitors as they provide many common features with the latter but they also provide advanced functionality. However, application servers are different from TP monitors (and traditional transaction middleware) in that they are able to function around the notion of transactional components, are easier in use and are less expensive than TP monitors (Linthicum, 1999a). However, they lack in performance and reliability compared to TP monitors and they are still immature. Like TP monitors, application servers do not do not support content or message transformation services without a lot of programming (Linthicum, 1999a).

Application servers support sharing and processing of application logic and provide connections to back-end resources such as enterprise resource planning systems, databases etc. Moreover, they provide the infrastructure for executing distributed applications together with technology integration capabilities and also provide user interface development mechanisms (Duke *et al.*, 1999). In addition, many vendors are attempting to incorporate functionality from message brokers to application servers (e.g. intelligent routing, transformation, messaging) and hence the latter are becoming more functional and flexible (Linthicum, 1999a). Application servers were developed for Internet based transactions and application development and new application servers tend to support EJB. Furthermore, they provide back-end integration capabilities, through a series of connectors provided by vendors such as SAP, BAAN etc (Linthicum, 1999a). Application servers need less time to build web-based applications. They provide the infrastructure support for executing distributed application with technology integration capabilities (Duke *et al.*, 1999). Application servers can communicate with DOT and increase their functionality (Linthicum, 1999a).

APPLICATION SERVERS	
Advantages	**Reference**
Support Component integration	Linthicum (1999b)
Support logic sharing and processing	Zahavi (1999)
Provide communication to packaged applications	Linthicum (2000)
Support e-business integration	Duke *et al.*(1999)
Provide communication to DOT	Zahavi (1999)
Flexible	Duke *et al.*(1999)
Disadvantages	**Reference**
Provide lower performance than TP monitors	Edwards and Newing (2000)
Provide lower reliability than TP monitors	Morgenthal and La Forge (2000)
Immature	Duke *et al.*(1999)
Support the translation layer in a complex and non flexible way	Linthicum (1999a)

Table B.8: Applications Servers - Advantages and Disadvantages

B.5 Distributed Objects Technology (DOT)

Coad and Yourdon (1990) report that the term "Object-Oriented" has been used in different ways within different disciplines such as Information Modelling and Object-Oriented Programming Languages. In addition, Object-Oriented (OO) is interpreted differently by different people ((Korson and McGrecor, 1990) and thus, there is confusion around this area. Therefore, to provide a common understanding it is meaningful to clarify what an Object, a Distributed Object and a Component is.

Coad and Yourdon (1990) define an object as *"an abstraction of something in a problematic domain, reflecting the capabilities of a system to keep information about it, interact with it, or both; an encapsulation of Attribute values and their executive Services."* Coad and Yourdon, (1990, p. 53). Simply, an object is the join of data, the methods and the functions to access and manipulate the data and are grouped into classes (Zahavi, 1999). Objects are run-time entities that encapsulate their state and behaviour (Korson and McGrecor, 1990;

Lycett, 1999) and support concepts such as: (a) Abstraction; (b) Encapsulation; (c) Class and Classification; (d) Hierarchy; (e) Polymorphism and, (f) Modularity (Coad and Yourdon, 1990; Zahavi, 1999). One of the aims of object technology is to provide reusable objects. However, this aim was elusive, as objects are too complex to manage across various applications and organisations and, they have also dependencies with applications that use the same libraries (Zahavi, 1999).

In contrast, distributed objects are more flexible and allow the sharing of data and methods through well-defined interfaces (Linthicum, 1999b). Distributed objects are not linked as part of applications but they communicate with other applications using their interfaces. Distributed objects allow developers to create portable objects that could run on a variety of servers and could communicating using a predefined and standard messaging interface (Linthicum, 1999a). However, not all the distributed objects support the concepts of: (a) Polymorphism; (b) Encapsulation and, (c) Inheritance (Zahavi, 1999).

Components are predefined pieces of application code that can be assembled into working application systems (Ruh *et al.*, 2000). Sparling (2000) defines components as *"a language neutral, independently implemented package of software services, delivered in an encapsulated and replaceable container, accessed via one or more published interfaces... A component is not platform constrained nor is application bound."* (Sparling, 2000), page 47). Components share similar characteristics of objects (Zahavi, 1999) but are not objects (Lycett, 1999). Lycett (1999) reports that components are higher-order abstractions that can be used to construct systems that are object oriented. Components can be local or distributed and part of them is based on object-oriented implementations (Mowbray and Zahavi, 1995). Components should be reusable, independent from the implementation approach and implementation language (Andrew, 1998) and, can be linked together with other components to create software applications (Zahavi, 1999).

Distributed Objects Technology (DOT) provides the ability to create Object-Oriented interfaces to existing or new applications that are accessible from any other application (Ruh *et al.*, 2000). It can also be used for the development of Component-Based systems. Distributed objects technology allows the sharing of data, application logic and provides a central clearinghouse for enterprise information (Linthicum, 1999b). Furthermore, it supports multi-tier architectures as well as the integration of multiple custom systems (Zahavi, 1999). As DOT matures, vendors are adding new features to their products to overcome drawbacks deal with scalability, interoperability and communication mechanisms. To achieve integration using DOT, changes to several applications should be done in order to share application methods with other distributed objects. Although DOT vendors claim that this is an easy matter, application code still have to be changed. As a result, distributed objects based integration is an invasive approach (Linthicum, 1999a).

Common Object Request Broker Architecture (CORBA), Component Object Model (COM), Distributed Component Object Model (DCOM) and Enterprise Java Beans (EJB) are the basic competing object technologies that are used to achieve integration (Linthicum, 1999a; Ruh *et al.*, 2000; Zahavi, 1999). These technologies are described in the following paragraphs.

B5.1 Common Object Request Broker Architecture (CORBA)

CORBA is a specification that software vendors can choose to incorporate it with their products (Ring and Ward-Dutton, 1999). This specification outlines the rules that vendors/developers should follow when creating a CORBA-compliant distributed object (Linthicum, 1999a). CORBA is open distributed object technology that enables remote object creation and remote object invocation. It is platform independent, provides standard object-oriented interfaces and specifies the interfaces that are used to access CORBA compliant software. It is based on the use of an object request broker (ORB)(Ruh *et al.*, 2000). ORB provides the communication infrastructure and it is responsible for locating and starting serves and, exchanging data between clients and servers. ORB provides platform independence and location transparency (Zahavi, 1999). CORBA, abstracts the communication level and system dependencies and allow applications to communicate with each other (Zahavi, 1999). It uses the Interface Definition Language (IDL) to define the interface between client and servers. The interface is the syntax of the operations that may be invoked on the server by the client as well as the data and the exceptions that may be exchanged.

CORBA was designed as an open standard to provide integration and interoperability in distributed systems (Rosen, 1998). It is available on more than 25 different platforms (Zahavi, 1999) and it is suitable for applications exist in a heterogeneous environment and especially when an organisation supports a variety of systems and platforms (e.g. UNIX, NT, mainframes etc) (Linthicum, 1999a; Ruh *et al.*, 2000). Moreover CORBA components can be used to support multiple programming languages (Zahavi, 1999). Many software vendors (e.g. BEA, IBM) support CORBA (Ring and Ward-Dutton, 1999) as well as many ERP vendors like (SAP, BAAN etc) integrate their packages with CORBA or Component Object Model (COM) (Zahavi, 1999). CORBA supports legacy integration, platform independence, location transparency, it is flexible and provides bindings to different languages (Zahavi, 1999).

COMMON OBJECT REQUEST BROKER ARCHITECTURE (CORBA)	
Advantages	**Reference**
Supports component integration	Zahavi (1999)
Supports legacy applications integration	Mowbray and Zahavi (1995)
Supports packaged applications integration	Ring and Ward-Dutton (1999)
Flexible mechanism	Wutka (2001)
Portable	Ruh *et al.*(2000)
Platform independent	Linthicum (1999a)
Provide real-rime integration	Ruh *et al.*(2000)
Supports components reusability	O'Callaghan (1999)
Supports heterogeneous back-end environments	Rosen (1998)
Disadvantages	**Reference**
Invasive mechanism	Ring and Ward-Dutton (1999)
Provides poor messaging capabilities	Zahavi (1999)
Offers low to medium scalability	Linthicum (2000)
Offers low to medium stability	Wutka (2001)
Immature	Linthicum (1999a)
Lack of provision for off the self ORB services	Ruh *et al.*(2000)
Does not support front-end environments (e.g. windows)	Rosen (1998)

Table B.9: CORBA - Advantages and Disadvantages

A basic disadvantage of CORBA technology is that it uses synchronous messaging communication and has poor capabilities for publish and subscribe messaging (Zahavi, 1999). However, the new version of CORBA3 seeks to provide asynchronous messaging but it is still immature. In addition, many CORBA implementations are not as stable as organisations require and have scalability limitations (Linthicum, 1999b). Another drawback derives from the fact that ORB vendors implement very few ORB services. This eliminates the advantages of the services, facilities and specifications because enterprises have to implement the missing services (Ruh *et al.*, 2000).

B5.2 Component Object Model (COM), COM+ and Distributed Component Object Model (DCOM)

COM is a distributed object standard provided by Microsoft for the Windows 95/98 and Windows NT platforms (Rosen, 1998). Although COM-enabled objects can support non-windows platforms (e.g. Unix, Mainframes), COM are more native to the Windows and are more homogeneous in nature (Linthicum, 1999a; Rosen, 1998). Like CORBA, COM provides the rules that developers should follow when creating COM-compliant distributed objects.

Initially, COM did not provide an ORB. As a result, a further development of COM includes the functionality of Microsoft ORB (Linthicum, 1999a) and led to COM+. COM+ intends to make COM programming easier, is easier to use (Andrew, 1998) and further enable significant component reuse (Ruh *et al.*, 2000). COM+ defines a standard set of types and makes components self-describing.

However, COM and COM+ were not as functional as should be without the ability to distribute COM-enabled ORBs. Thus, Microsoft developed DCOM as part of Windows operating system (Windows 98, Windows NT, and Windows 2000). DCOM allows COM enabled application to locate and use remote COM-enabled ORB and find and invoke the service it requires (Linthicum, 1999a; Zahavi, 1999). DCOM is compatible with existing COM-enabled development. DCOM is windows-bound ORB, is still immature and does not provide the functionality and the performance of CORBA. Like CORBA, DCOM supports more synchronous communication models and it is immature to support asynchronous types of communications. COM and DCOM support multiple languages but they differ with CORBA in their approach to support these languages. Other differences between DCOM and CORBA include differences in data types, exceptions, services, life cycle etc. (Rosen, 1998; Zahavi, 1999)

DISTRIBUTED COMPONENT OBJECT MODEL (DCOM)	
Advantages	**Reference**
Supports components reuse	Ruh *et al.*(2000)
Supports packaged applications integration	Wijegunarate and Fernandez (1998)
Provides real – time integration capabilities	Zahavi (1999)
Easy to use	Andrew (1998)
Supports Windows environments	Linthicum (2000)
Disadvantages	**Reference**
Does not support efficiently back-end environments	Linthicum (1999a)
Immature to support asynchronous communication models	Rosen (1998)
Provides less functionality and performance than CORBA	Zahavi (1999)
Invasive mechanism	Rosen (1998)

Table B.10: DCOM - Advantages and Disadvantages

B5.3 Enterprise Java Beans (EJB)

Java Beans are components written in Java and expose information about their methods, attributes and events through classes (Morgenthal, 1998). Java Beans can send and receive events and communicate with each other (Zahavi, 1999). Enterprise Java Beans (EJB) define a model for the development of reusable Java server components (Ruh *et al.*, 2000) and is similar to COM and CORBA architectures (Linthicum, 1999a). It supports multi-tier distributed objects application development and provides a set of enterprise component interfaces (APIs) (Morgenthal, 1998; Ruh *et al.*, 2000). EJB component model is strongly transactional and can support asynchronous communications and publish/subscribe services (Ruh *et al.*, 2000). EJB is communication protocol independent, it is easy to use (Andrew, 1998) and, provides portability and simpler development (Zahavi, 1999). Application servers can be adapted to support EJB by adding support for services defined in EJB specification (Linthicum, 1999a). Enterprise Java Beans are the best technology to support homogeneous Java environments. Recently, there is an attempt for the development of mappings between CORBA and EJB and there is a convergence of their components. Therefore, CORBA and EJB are much more interoperable than CORBA and DCOM (Zahavi, 1999). However, COM and DCOM are more appropriate for front-end environment (e.g. Windows) and CORBA with EJB for back-end systems.

ENTERPRISE JAVA BEANS (EJB)	
Advantages	**Reference**
Supports Java components reusability	Monson-Haefel (2000)
Protocol independent	Wutka (2001)
Easy to use	Andrew (1998)
Portable	Zahavi (1999)
Supports Java homogeneous environments	Roman *et al.*(2001)
Supports mappings to CORBA	Zahavi (1999)
Supports back-end systems	Roman *et al.*(2001)
Disadvantages	**Reference**
Can not support front – end systems	Ruh *et al.*(2000)
Invasive	Linthicum (1999a)

Table B.11: EJB - Advantages and Disadvantages

B.6. Interface Oriented Technologies

As reported above, integration can be message or interface based. Interface based integration requires the specification of well-defined interfaces that describe the actions that an application can perform (Ruh *et al.*, 2000). Interfaces are means by which users, application or software components can interact with a given application or component (Zahavi, 1999) and they are associated with an application. In contrast, messages are not associated with applications require more processing to decode and errors are discovered later in the development process than an interface (Ruh *et al.*, 2000). Interfaces are externally visible and usable by any application and they are also easier to reuse and maintain than messages. However, they may be more difficult to change and extend. Interfaces can reduce both the time of integration and the time of maintenance applications (Klasell and Dudgeon, 1998). They can be classified into: (a) User Interfaces; (b) Application Programming Interfaces (APIs) and, (c) Adapters (or Connectors).

B.6.1. User Interfaces and Screen Scrappers (or Wrappers)

In many applications such as legacy systems, the user interface is the only available mechanism to access data, logic and processes (Andrew, 1998; Van Den Heuvel *et al.*, 1999). A category of tools called Wrappers is used to reuse useful business rules and data from existing applications and/or to ensure integrity between new and existing systems (Butler Group, 1998). Screen Wrappers (or Scrappers) is a type of wrappers that are used to encapsulate (extract) data from user interfaces and transform them into raw data (screens as data) or objects (screens as objects) (Linthicum, 1999a). In doing so, they use mapping techniques to map the user's interface information to raw data or objects. Hence, screen wrappers should be used in cases that user interfaces rarely change (Linthicum, 1999a). Wrappers are also expose interfaces over legacy transactions and provide metadata descriptions of legacy systems (Robertson, 1997). Both methods (screens-as-data and screens-as-objects) are non-invasive (Andrew, 1998; Noffsinger *et al.*, 1998). Screens-as-data wrapping is simpler than the screens-as-objects method. However, by using screens as data, the application methods that act on the information extracted from the user interface are not encapsulated. The screens-as-objects method requires translating the data extracted from user interface to an application object (e.g. CORBA, COM or Java) which requires adding the application methods needed to interact on the data (Linthicum, 1999a). As a result, user interface information is transformed into a set of objects that can be processed by DOT. Thus, the screens-as-objects method is more appropriate for EAI. In addition data objects reduce maintenance efforts and increase scalability (Morgenthal and La Forge, 2000).

Screen Wrappers	
Advantages	**Reference**
Reduce time of integration	Klasell and Dudgeon (1998)
Support translation layer	Linthicum (1999a)
Non invasive	Noffsinger *et al.*(1998)
Reduce time of maintenance	Klasell and Dudgeon (1998)
Support the integration of legacy systems	Robertson (1997)
Increase scalability	Morgenthal and La Forge (2000)
Reduce maintenance effort	Morgenthal and La Forge (2000)
Support Reusability	Butler Group (1998)
Disadvantages	**Reference**
Face maintenance problems when screens are changed	Andrew (1998)
Does not support process automation layer	Wijegunarate and Fernandez (1998)
Does not support transportation layer	Linthicum (2000)

Table B.12: Screen Wrappers - Advantages and Disadvantages

B6.2 Application Programming Interfaces (APIs)

Application Programming Interfaces (API) is a mechanism provided by an application to access its functionality or data (Ruh *et al.*, 2000). Linthicum (2000) defines an API as *"well-defined mechanisms that are built to connect to some sort of recourse, such an application server, middleware layer, or database. ... APIs allow developers to invoke the services of these entities on order to obtain some value."* Linthicum, (1999a, p. 39). APIs can communicate with DOT such as CORBA and DCOM and they are portable to other applications (Zahavi, 1999).

Nowadays, the majority of packaged solutions (e.g. ERP systems etc) use APIs to facilitate the communication and the integration of these applications with others. As a result, ERP systems are attempting to function like distributed object solutions by supporting APIs that provide access to their data, objects, services and processes. APIs can be used for accessing data in real time, and support reusability (Linthicum, 1999a). However, they are extremely complex and may be changed in the next version of the software product. This means, that applications that are integrated using APIs can be broken if one application makes a change that is not immediately recognised across all applications that communicate with it (Morgenthal and La Forge, 2000). Recently, there is a trend for standardising enterprise wide APIs, which results in reducing maintenance cost and efforts.

APPLICATION PROGRAMMING INTERFACES (APIs)	
Advantages	**Reference**
Portable mechanisms	Mende (2000)
Support translation layer	Angeli *et al.*(2000)
Reduce time of integration	Klasell and Dudgeon (1998)
Support reusability	Linthicum (1999a)
Communicate with DOT	Zahavi (1999)
Non-invasive	Ruh *et al* (2000)
Reduce time of maintenance	Klasell and Dudgeon (1998)
Disadvantages	**Reference**
High Complexity	Morgenthal and La Forge (2000)
Does not support process automation layer	Wijegunarate and Fernandez (1998)
Does not support transportation layer	White *et al.*(1999)

Table B.13: APIs - Advantages and Disadvantages

B6.3 Adapters or Connectors

The concept of an adapter (or connector) is an extension of DBMS drivers (Ring and Ward-Dutton, 1999). Ruh *et al.*(2000) define adapters as "*software components that enable interconnection between applications. They provide the connection to each application (adapter) and the 'pipe' between the connectors.*" Ruh *et al,* (2000, p. 149). An adapter is a set of libraries that map differences between two distinct interfaces, hides the complexities of interfaces (Linthicum, 1999a) and performs the extraction, translation and input steps of an integration (Klasell and Dudgeon, 1998). Adapters can be used as translators that plug into application APIs to present standardise data or message interfaces to the connectivity transformation and process management services of an EAI solution. Moreover, adapters offer more capabilities than APIs (Ring and Ward-Dutton, 1999) and support error handling, marshalling and unmarshalling of data from messages or objects, transformation of data into formats that are acceptable to the target application (Ring and Ward-Dutton, 1999). They add more value to EAI solutions by supplying metadata that describes resource behaviour to developers (Ruh *et al.*, 2000). In addition Adapters are transparent, hide the complexity and achieve reusability (Linthicum, 1999a). They are becoming more sophisticated and are getting smarter by incorporating intelligence.

ADAPTERS or CONNECTORS	
Advantages	**Reference**
Support transformation layer	Ring and Ward-Dutton (1999)
Non Invasive method	Roman *et al.*(2001)
Achieve reusability	Linthicum (1999a)
Support data, messages and objects integration	Ruh *et al.*(2000)
Hide the implementation complexity	Morgenthal (1999)
Reduce time of integration	Wutka (2001)
Reduce time of maintenance	Klasell and Dudgeon (1998)
Disadvantages	**Reference**
Does not support process automation layer	Wijegunarate and Fernandez (1998)
Does not support transportation layer	Linthicum (2000)

Table B.14: Adapters - Advantages and Disadvantages

C

APPENDIX C: Interview Agenda

This questionnaire is divided into 3 parts

The questionnaire aims to address the following issues:

- To obtain general company information
- To obtain technical information
- To identify business information (e.g. benefits, barriers and costs associated with application integration)

Name:			
Position:			
Company Name:			
Address:			
Telephone:		**Fax:**	
e-mail:			

Sections

Section A – General Company Information

Section B – Technical Information

Section C – Business Information

Section A – General Company Information

A.1 How many people are employed by your organisation? [ ]

A.2 How many subsidiaries does your organisation have? [ ]

A.3 Is your organisation a multinational company?

Yes	
No	

If yes, please specify in how many countries does your organisation operate:

A.4 What is the current turnover? [ ]

A.5 What are the key businesses of your organisation?

A.6 How many customers do you have (approximately): [ ]

Section B – Technical Information

B.1 How is your IT infrastructure organised? Is there any central integrated infrastructure or each subsidiary (or department) has its own infrastructure?

Please explain:

B.2 How many information systems do exist in your organisation? Please specify types of system (e.g. packaged, custom, ebusiness), platform (e.g. AS/400), numbers:

Type of System	Platform	Amount

B.3 What problems did you have before adopting an Enterprise Application Integration (EAI) solution (e.g. data accuracy)?

Please explain:

B.4 Which adoption approach did you follow when integrating your organisation:

(a) Data Centric Approach:

(b) Process Centric Approach:

B.5 How is the big picture of the integrated IT infrastructure look like? Please draw a figure of the overall EAI solution:

B.6 Could you specify the duration in the following table:

Phase	**Duration**	**Start – End**
System Design		
Business Process Reengineering		
Implementation		
Testing		
Other:		

B.7 Could you specify how many information systems have been phased out due to EAI adoption?

Type of System	**Platform**	**Amount**

B.8 What types of systems did you integrate?

Type of System	Platforms	Amount
Custom-to-custom		
Custom-to-packaged		
Custom-to-ebusiness		
Packaged-to-Packaged		
Packaged-to- ebusiness		
Custom-to-packaged-to-ebusiness		

B.9 Could you describe the integration phases?

Phase	Description

B.10 Have you evaluated EAI tools before adopting one?

Yes	
No	

B.11 How important are the following requirements when integrating your applications? Please tick

Integration Requirements	1 (Not Important)	3	5 (Very Important)
Maintainability			
Flexibility			
Scalability			
Portability			
Reusability			
Product Maturity			
Technology Maturity			
Complexity			
Non-invasive			
High Performance			
Real Time Integration			
Mainframe Compatibility			
Non-Mainframe Compatibility			
Support of Data Integration			
Support of Objects Integration			
Support of Process Integration			
Support of Transportation Layer			
Support of Transformation Layer			
Support of Process Automation Layer			
Support of Custom-to-Custom Application Integration			
Support of Custom-to-Package Application Integration			
Support of Custom-to-ebusiness Application Integration			
Support of Packaged-to- Packaged Application Integration			
Support of Packaged -to-ebusiness Application Integration			
Support of ebusiness-to- ebusiness Application Integration			
Support of Custom-to-Packaged-to- ebusiness Application Integration			
Other:			
Other:			
Other:			
Other:			
Other:			
Other:			
Other:			
Other:			
Other:			

B.12 Application elements are those elements that are used for the integration of two applications and include data, objects and processes. Below identify how important are the following technologies for the integration of those application elements. Please tick.

	Application Elements								
	Data			**Objects**			**Process**		
Integration Technologies	**1**	**3**	**5**	**1**	**3**	**5**	**1**	**3**	**5**
ODBC									
JDBC									
RPC									
MOM									
Message Broker									
XML									
TPM									
Application Serves									
CORBA									
DCOM / COM									
EJB									
Screen wrapper									
APIs									
Adapters									

B.13 Integration is taking place at three layers namely (a) transportation; (b) transformation and (c) process automation. The first layer is used for transmitting application elements from source application to target. The second layer translates application elements from source application format to target's where the third layer controls and integrates the whole integration process. Below, define how important are the following technologies for those integration layers. Please tick.

	Application Elements								
	Transportation			**Transformation**			**Process Automation**		
Integration Technologies	**1**	**3**	**5**	**1**	**3**	**5**	**1**	**3**	**5**
ODBC									
JDBC									
RPC									
MOM									
Message Broker									
XML									
TPM									
Application Serves									
CORBA									
DCOM / COM									
EJB									
Screen wrapper									
APIs									
Adapters									

B.14 How significant are these technologies in integrating the system types presented in the table below? Please tick.

	System Types																				
	Custom-to-Custom			Custom-to-Packaged			Custom-to-ebusiness			Packaged-to-packaged			Packaged-to-ebusiness			Ebusiness-to-ebusiness			Custom-to-Packaged-to-ebusiness		
Integration Technologies	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5
ODBC																					
JDBC																					
RPC																					
MOM																					
Message Broker																					
XML																					
TPM																					
Application Serves																					
CORBA																					
DCOM / COM																					
EJB																					
Screen wrapper																					
APIs																					
Adapters																					

B.15 How important are the integration technologies in meeting the following integration requirements? Please tick.

	Integration Requirements																	
	Maintainability			Flexibility			Scalability			Portability			Reusability			Maturity		
Integration Technologies	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5
ODBC																		
JDBC																		
RPC																		
MOM																		
Message Broker																		
XML																		
TPM																		
Application Serves																		
CORBA																		
DCOM / COM																		
EJB																		
Screen wrapper																		
APIs																		
Adapters																		

Requirement	Description
Maintainability	It refers to the capability of software applications to allow changes without causing problems to other components or systems. In integration area, technologies should lead to the development of solutions that could be easily maintained.
Flexibility	Flexibility makes a system or a process able to respond to change in the system's environment
Scalability	Scalability describes the ability of an Information System to provide high performance as greater demands are placed upon it, through the addition of extra computing power
Portability	Portability allows a software solution developed for one platform to run on an entirely different platform
Reusability	Reusability refers to the capability of using existing components or software solutions to build new applications
Maturity	Maturity shows whether an integration technology is mature or not. The more mature an integration technology is the better

B.16 How important are the integration technologies in meeting the following integration requirements? Please tick.

	Integration Requirements																	
	Low Complexity			Non-invasive			High Performance			Real-Time			Mainframe Compatible			Non-Mainframe Compatible		
Integration Technologies	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5
ODBC																		
JDBC																		
RPC																		
MOM																		
Message Broker																		
XML																		
TPM																		
Application Serves																		
CORBA																		
DCOM / COM																		
EJB																		
Screen wrapper																		
APIs																		
Adapters																		

Requirement	Description
Low Complexity	Complexity describes whether an integration technology leads to complex or simple solutions. Often, complex integration solutions are not preferred as they increase development and maintenance costs.
Non-invasive	A technology is non-invasive when requires no or limited changes to existing applications code to achieve integration.
High Performance	In some cases integration technologies achieve integration but the performance of the overall solution could be low. The higher the performance is the better.
Real-Time	Real Time requirement refers to the capability of integration technologies to support transactions that require up to the second data latency. Data latency defines how current the information needs to be.
Mainframe Compatible	This describes whether a technology supports the integration of mainframe based applications
Non-Mainframe Compatible	This describes whether a technology achieves the integration of systems that run on non-mainframe environments.

Section C – Business Information

C.1 Who initiated the idea for adopting an EAI solution?

C.2 What were the main motivations for adopting an EAI solution?

C.3 What business problems did your organisation face before integrating applications?

C.4 What was the impact from the adoption of EAI? Please explain:

Impact on organisation:

Impact on employees:

C.5 What was the overall cost for the EAI solution? Please explain:

C.6 What are the main costs associated with the adoption of EAI in your organisation?

Application Integration Costs	1 (Not Important)	3	5 (Very Important)
Hardware costs			
Software costs			
Development costs			
Maintenance costs			
Consultancy costs			
Employees training			
Changing employees culture			
Management efforts			
Business Process re-engineering			
Organisational restructuring			
Covert resistance			
Strategy redesign			
Other:			
Other:			
Other:			
Other:			
Other:			

C.7 What benefits are derived from EAI adoption in your organisation?

Application Integration Benefits	1 (Not Important)	3	5 (Very Important)
Provides more understanding and control of processes			
Improves management and supports decision making			
Results in more organised business processes			
Allow organisations to do business more effectively			
Improves planning in supply chain management			
Increases collaboration among partners			
Reduces lost sales			
Increases productivity			
Increases performance			
Achieves customer satisfaction			
Results in reusable systems, components and data			
Reduces redundancy of applications, data and tasks			
Reduces cost			
Achieves return on investment			
Faster and cheaper implementation than bespoke solutions			
Increases flexibility			
Quicker response to change			
Offers interfaces-standardisation			
Provides flexible, maintainable and manageable solutions			
Results in reliable data			
Process and systems scalability			
Provides portability			
Reduces development risks			
Achieves non-invasive solutions			
Achieves process integration			
Increases data analysis			
Improves data quality			
Supports efficient data sharing			
Other:			
Other:			
Other:			

C.8 What are the barriers to EAI adoption in your organisation?

Application Integration Barriers	**1** (Not Important)	**3**	**5** (Very Important)
Politics and political impact (e.g. who controls the processes)			
Resistance to change			
No single EAI product solves all integration problems			
No time for training employees on integration technologies			
Extra cost for redesign and change business structure, processes			
Lack of employees with EAI skills			
Cultural issues			
EAI has a high cost			
High complexity in understanding the processes and systems in order to redesign and integrate them			
Earlier approaches on EAI had proved problematic			
Complexity of business processes			
Other:			
Other:			
Other:			
Other:			
Other:			
Other:			