

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

An Investigation of System Integrations and XML Applications within a NZ Government Agency

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Information Systems

at

Massey University, New Zealand

by

Steven Li

2009

ABSTRACT

With the evolution of Information Technology, especially the Internet, system integration is becoming a common way to expand IT systems within and beyond an enterprise network. Although system integration is becoming more and more common within large organizations, however, the literature review had found IS research in this area had not been sufficient, especially for the development of integration solutions within large organizations. It has made research like this one conducted within a large NZ government agency necessary. Four system integration projects were selected and studied using case study research methodology. The case study was designed and conducted using guidelines mainly from the well-known R. K. Yin's (2002) "Case Study Research" book.

The research was set to seek answers for a series of research questions, which were related to requirements of system integration and challenges for solution development. Special attention had been given to XML applications, as system integration and XML were found to be coupled in many system integrations and frameworks during the literature review. Data were first gathered from all four projects one by one, and then the bulk of analysis was done on the summarized data. Various analysis methods including chain-of-evidence, root-cause-analysis and pattern-matching were adopted. The principles of interpretive research proposed by Klein and Myers (1999) and triangulation were observed.

In conclusions, a set of models have been derived from the research, namely a model for clarifying integration requirements; a model for integration solution architecture; a model for integration development life cycle and a model of critical success factor for integration projects. A development framework for small to medium size integration projects has also been proposed based on the models. The research also found XML application indeed would play an important role for system integration; the critical success factors for XML application included suitable development tools, development skills and methodologies.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my supervisors, Dr. Alexei Tretiakov and Dr. Dick Whiddett, for the invaluable guidance and generous help given throughout this research project. This research would never have been possible without their guidance and support.

I am deeply grateful to my employer, a New Zealand government agency, for supporting me during my entire study in Massey University. I am especially grateful to the people from the IT team of the agency for allowing and assisting me to conduct this research project. Special thanks also go to people who have provided comment and feedback for the draft report. Without the generous help from all these people, this research would not have been possible to complete.

I am as ever, especially indebted to my family for their love and support throughout my study. They are the ultimate source for my energy and study effort. I wish to express my wholehearted appreciation and gratitude to them all.

Table of Contents

Chapter 1 Introduction	1
1.1 Research Background.....	1
1.2 Research Objectives	1
1.3 Significance	1
Chapter 2 Literature Review	5
2.1. Introduction	5
2.2 System Integration.....	6
2.3 Enterprise Application Integration	27
2.4. Development of System Integration	46
2.5. Perspectives of XML Application	65
2.7. Review Conclusions	88
Chapter 3 Research Proposal and Method	91
3.1 Research Questions	91
3.2 Research Methodology	94
3.3 Case Study Protocol	102
Chapter 4 Case Summary	119
4.1 Case 1: GIS Integration	119
4.2 Case 2: DM Integration	124
4.3 Case 3: Office & Person Data Sharing	128
4.4 Case 4: Corporate Directory	132
Chapter 5 Case Study Findings and Analysis	137
5.1 Summary and Discussion of all Cases.....	137
5.2 Analysis of the Findings	163
5.3 Findings for the Research Questions	181
Chapter 6 Conclusions and Discussions	193
6.1 Conclusions	193
6.2 Research Contributions	204
6.3 Research Limitations	206
6.4 Future Research.....	207
Reference	209
Appendix A: Case 1 Data Collection & Preliminary Analysis	213
Appendix B: Case 2 Data Collection & Preliminary Analysis	247
Appendix C: Case 3 Data Collection & Preliminary Analysis	277
Appendix D: Case 4 Data Collection & Preliminary Analysis	301
Appendix E: Review of IS Research Method	321

List of Figures

Figure 2. 1 Diagram by E. Rahm, P.A. Bernstein: A survey of approaches to automatic schema matching, 2001	11
Figure 2. 2 Diagram by Jinyoul Lee, Keng Siau, and Soongoo Hong (2003)	29
Figure 2. 4 Figures by Marinus Themistocleous, Zahir Irani, Robert M. O’Keefe and Ray Paul (2001)	34
Figure 2. 5 Figure by Vishnu S. Pendyala, Simon S.Y. Shim, Jerry Z. Gao (2003)	39
Figure 6. 1 Integration Requirement Model	194
Figure 6. 2 Integration Solution Design Model	195
Figure 6. 3 Integration Development Model	197
Figure 6. 4 Critical Success Factor (CSF) Model	198
Figure 6. 5 Model Based Enterprise Integration Development (MB – EAID) Framework	202
Figure A. 1 DOCgis Main Screen	216
Figure A. 2 DOCgis Map Screen	217
Figure A. 3 DOCgis Spatial Query	217
Figure A. 4 DOCgis Query Result Sample	217
Figure A. 5 DOCgis Centred National Enterprise GIS Architecture	219
Figure A. 6 GIS Viewer Main Screen	222
Figure A. 7 GIS Viewer Spatial Query on User Defined Area	223
Figure A. 8 GIS Viewer Spatial Query on Existing Defined Area	223
Figure A. 9 GIS Viewer Spatial Query Report Result Sample	224
Figure A. 10 GIS Viewer Map Sample	224
Figure A. 11 Permissions/GIS Integration Screen Sample 1	225
Figure A. 12 Permissions/GIS Integration Screen Sample 2	225
Figure A. 13 GIS Viewer System Architecture	226
Figure B. 1 Document Creation Screen 1	250
Figure B. 2 Document Creation Screen 2	250
Figure B. 3 Document Creation Screen 3	251
Figure B. 4 Document Integration Screen	252
Figure B. 5 DM Integration Architecture 1	253
Figure B. 6 DM Integration Architecture 2	254
Figure B. 7 DM Integration Revised Architecture 1	256
Figure B. 8 DM Integration Revised Architecture 2	257
Figure B. 9 DM Integration Second Revised Architecture 2	259
Figure B. 10 DM Integration Land Register Sample Screen	259
Figure C. 1 PerOrg Prototype Screen 1	280
Figure C. 2 PerOrg System Architecture	281
Figure C. 3 PerOrg with Common Data Proposal	282
Figure C. 4 Office & Person Common Database ERD	284
Figure D. 1 Corporate Directory Data Flow Diagram	304
Figure E. 1 Figure by M. R. de Villiers (2005)	327

List of Tables

Table 2. 1 Summary table from “Business-to-business e-commerce frameworks” by Shim, S.S.Y., Pendyala, V.S.; Sundaram, M., Gao, J.Z. (2000)	26
Table 2. 2 Table 1 by Nina Reiersgaard, Hilde Salvesen, Stig Nordheim, Tero Päiväranta (2005)	53
Table 2. 3 Table 2 by Nina Reiersgaard, Hilde Salvesen, Stig Nordheim, Tero Päiväranta (2005)	53
Table 2. 4 Table 3 by Nina Reiersgaard, Hilde Salvesen, Stig Nordheim, Tero Päiväranta (2005)	54
Table 2. 5 Table 1 by Richard Berntsson-Svensson, Aybüke Aurum (2006).....	58
Table 2. 6 Table 2 by Richard Berntsson-Svensson, Aybüke Aurum (2006).....	59
Table 2. 7 Table by Wing Lam (2005).....	60
Table 2. 8 Table by Zaitun A. B. and Mashkuri Yaacob (2000)	62
Table 2. 9 Quality Assurance Measures	115
Table E. 1 Table by Robert L. Glass, V. Ramesh, Iris Vessey (2004)	324
Table E. 2 Table by Yin (2002).....	331

Chapter 1 Introduction

1.1 Research Background

This is a thesis paper for Master of Information System. The research was to investigate system integrations and XML applications in a government agency. The student of this project had worked for the government agency for over six years as a software developer. Both system integrations and XML applications are important for the organization and are often encountered as part of the daily work. The research project was selected not only because there were a large repository of research data available, more important, the outcome of the research would be significant for academic and very useful for the agency.

1.2 Research Objectives

The followings are objectives for the research project:

- 1) To identify the core issues for system integration from system development perspective within a large organization such as a government agency.***
- 2) To identify techniques to resolve these issues, with particular interest in XML application.***

1.3 Significance

Reasons for choosing the project objectives are described below.

- 1) System integration is an important subject. As organizations have built or acquired more and more information systems, they constantly face the question of whether to build more new systems or to make use of those existing systems for new requirements or new functions in the ever-changing business and organization. As it was argued by Erik G. Nilsson Else K. Nordhagen Gro Oftedal (1990), systems integration has become one of the hottest buzzwords for recent years. Because of its importance, the term tends to be used for anything that is not totally isolated. But not many cases have actually carried out practical system integration.
- 2) As it was summarized in the literature review, there are many different types of system integrations, because there are so many types of systems for any large organization, such as government agencies. By reviewing a large amount of literatures, many theories were found focused on general integration across

different systems and organizations, especially for B2B. However, the challenge of developing effective and efficient integration system within an organization was not found to be discussed in many papers. Such challenge cannot be adequately addressed with general theories because systems within an organization would have many unique characteristics. A focused study on this topic would be much needed and potentially provide great benefits.

- 3) During the literature review, many papers were found arguing that system development and integration were greatly influenced by human behaviours, management of the organization and politics. In a paper by Lam (2005), the author categorized critical success factors (CFS) of enterprise application integration (EAI) into 4 areas including business, organization, technology and project (BOTP model). Most technology factors were placed low in the priority list. In many other papers, technical issues also tend to be placed in much lower priority than management ones, if they are not totally ignored. In the Standish Group (2004) Chaos Report, skill resources was place in 8th in the top 10 success factors for IT project, well behind things like user involvement, management support, business objectives. However, technology is one of the core elements for any IT project and it makes IT systems unique and powerful. The researcher believed technical issues would require more attentions in IS research.
- 4) From the literature review, many XML applications were found in document sharing and publishing areas. The review also identified there was a trend of using XML data format and XML web service applications. Although the researcher believed XML applications especially XML web service would play a major role in system integration development, such assumption was not found to be proven true or false from the review. An investigation into the areas with actual cases would generate results for providing guidelines for future projects.
- 5) Government sectors are of significant part of IT industry. Many research papers were found discussing IT topics within government agency environment, including system integration and XML applications. System integrations including complex data sharing and functional integration are common in a large government agency. However, a lot of XML discussions for government agencies are focused on data standards and documents, which are influenced largely by policy and politics. There was a lack of research of practical applications of XML technology for the purpose of system development and integration. It was believed opportunities for

using XML in system integration within government agencies are still to be explored.

- 6) Development of system integration would have a unique set of characteristics different from general system development, particularly when integration is among systems within an organization. In the literature review, many development methodologies and theories were found created for general system development and integrations, such as framework for data sharing across multiple platforms by Janusz R. Getta (2006) and behaviour-based integration of autonomous object-oriented systems by Markus Stumptner, Michael Schrefl, Georg Grossmann (2004). Very few papers were found addressing issues concerning those various but relatively small integration projects within an organization. Much of the integration development would be within the organization and would face similar challenges. A research in this area would be much needed.
- 7) Any insight for the research objectives discovered in the research will be very useful for the organization. The researcher has many years of system development experience in a large government agency with a large number of systems. He had found many of system development within the agency involved some sort of system integrations. However they would be of various types and scale, in many occasions, they would not be addressed with any of current integration theories or integration tools on the market. Many issues have arisen and made these integrations very challenging and put the developments in risk if they were not treated properly.

Chapter 2 Literature Review

2.1. Introduction

System integration is the linking together of the subsystems physically or functionally. It has gone far beyond the basic networking of computer systems. Technologies of networking and communications are underlying foundations for all system integrations. As these technologies advance further and further such as the evolution of Internet, they have become the main driving force for the development of system integration, along with new business requirements. System integration is about achieving new goals and adding values to existing systems through the interaction among them. With the fast expansion of information systems in almost every organization and the availability of Internet, new and complicate system integrations are becoming feasible and desirable. More organizations would evaluate their existing systems before any new major IT investment. Development and implementation of system integrations have often become a preferred option.

The Extensible Markup Language (XML) is a subset of Standard Generalized Markup Language (SGML) for data publishing. XML would be regarded as a language to describe data it carries within the document, since its introduction in 1996 by the XML Working Group (originally known as the SGML Editorial Review Board), out of its original goal for enabling generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML, XML has been developed as a platform-independent and future proven format for storing and passing information, especially in the Internet environment. Dr Charles Goldfarb, who was personally involved in its invention, claimed XML to be “*the holy grail of computing, solving the problem of universal data interchange between dissimilar systems.*”

XML applications would bring many benefits to system integration, and more and more system integrations have made use of XML. Numerous integration theories and frameworks have been proposed and based on XML or its application, such as web service. Most of them were targeted for generic and cross organization situations such as B2B over Internet. However, system integrations within the organization boundary are also becoming one of the hot topics and the implementations of them would be

seen more often. The research and development in this area are becoming a popular IS research topic and important for the industry.

2.2 System Integration

2.2.1 Overview

Systems integration would include a huge number of topics and it means different thing to different people. In an early paper by Nilsson, E.G.; Nordhagen, E.K.; Oftedal, G. (1990), they argued that there were four main aspects of system integration, each focuses one aspect of integration and could be addressed independently off the other. They are integration technology, integration architecture, semantic integration, and user integration. The followings are the definition according to their theory.

Integration Technology

It focuses on the underlying techniques and mechanism of the integration. In the modern software development world, there are so many different techniques in different layers of the component structure. Some of the examples are EDI, object broker, TCP/IP, Web Service. They are essential for any integration and the selection of them can have big impact on the integration outcome.

Integration Architecture

It focuses on software design aspect of the integration. It should aim for achieving the result in a safe, easy and efficient way. New types of system architecture are coming at the time. Some of the examples are Client Server, Component object model, DOCM, Message Queue and Service Oriented Architecture.

Semantic Integration

It focused on the semantic content of data in different systems. Data sharing and database schema are the central part of many integration. Data from different systems, especially from different vendors, tends to be very different. It is crucial semantic integration can be implemented properly in those situations.

User Integration

It focuses on the systems from the end user point of view. Many system integrations are aimed for providing users an integrated interface to all underlying systems. Examples of those are Intranet portal, embedded objects and functions, and task oriented system. It is crucial for creating user-friendly environment and improving efficiency for the organization information systems.

The category by Nilsson, E.G.; Nordhagen, E.K.; Oftedal, G. (1990) was mainly based on the development (technology and target area) domain of system integration. As they also, said, all the aspects need to be considered for any system integration, as it would be complicated and difficult.

However, there are many other aspects of system integrations and different ways to approach the topic for research. For example, system integrations can be separated according to whether they are for new or old systems. Because most new systems were built with integration in mind and modern software development techniques are more integration friendly than those in the past. When talking about integration, we need to consider whether it is for legacy systems or new systems. Because few legacy systems are integration friendly; and new systems can be developed with integration friendly techniques.

In summary, system integration can be a very general term in information technology. Different levels of integration may be achieved in the different technology and functional areas. An integration of a certain set of systems may also vary from different points of view including business and technology. For example, it may have a high level of semantic integration, but only low-level integration technology. For the purpose of our research, we adopt a definition for system integration as following, *system integration is a process of linking different sub-systems or software applications together functionally, especially for providing a particular result or achieving a particular goal.*

2.2.2 Current Challenges

2.2.2.1 Overview

In the past ten years, as Internet and other computer technologies have evolved faster and faster, it has brought even more challenges and excitements to system integration researchers and practitioners. Many system integrations have been achieved and brought great benefit to users, especially in B2B commercial applications. The review had found a lot of integration theories and frameworks in commercial and academic world. For example, in a paper by Markus Stumptner, Michael Schrefl, Georg Grossmann (2004), they proposed a behaviour-based integration. They argued because

modern software systems tend to be less coupled, more distributed and relying on the Internet and web service, the traditional and rigid integration in data and structure are becoming less common. Their behaviour-based integration used meta-class objects to describe behaviours of objects for integration. By following rules and checking object behaviours, users can create integrations dynamically. Their theory was then focused on database and system (business) process. By reviewing a large amount of literatures, it was concluded that there were a few areas that had attracted the most attentions and effort in the recent years in the integration technology domain. They are data integration, functional integration and workflow integration.

2.2.2.2 Data integration

Overview

In a paper by Maurizio Lenzerini (2002), he defined data integration as the process of combining data residing at different sources and providing the user with a unified view of these data. He also argued that a typical enterprise would have multiple information systems and multiple databases. With the availability of Internet and latest communication technologies, it is becoming more and more common for information systems to be built on various data sources over wider and wider networks using Internet. As it was argued by Janusz R. Getta (2006), the central problem for the development of global information systems was to establish efficient data integration over the wide area networks. Many papers have been founded in the review on data integration, because data tends to be the foundation of any system development; and there are so many challenges and new technologies arising in the past few years. The review has found the challenge of data integration was often argued to be one of those most common and most important issues in the IT industry. As it was argued by Alon Halevy, Anand Rajaraman, Joann Ordille (2006), data integration would be very challenging because it needed to query across multiple autonomous and heterogeneous data sources. Especially in a large enterprise system that owns or uses a multitude of data sources, data integration is crucial. For example, in large-scale scientific projects, where data sets are being produced independently by multiple researchers, integration of all these data provides the foundation for any further progress for the research. In government agencies each with their own data sources, data integration is crucial for better cooperation and joint tasks. For building Internet search and web portals, data

integration is the only way for accessing numerous structured data sources over the Internet.

The followings are some of the opinions from various papers and sources:

Peer-to-Peer data (P2P) integration

There are many approaches and theories trying to define and come up with generic solutions for data integration. An early Peer-to-Peer data (P2P) integration theory was proposed by Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Riccardo Rosati (2004). In simple words, it is to establish mapping among the peer schemas for the data integration. It was argued that such approach is superior with respect to the three central properties of data integration: modularity, generality, and decidability. Modularity is that the integration with other source should not cause change to its own. Generality is about how free the integration connection can be placed among the sources. Decidability is about the ability of answering queries and quality assurance of those answers. In the paper, they proposed a new semantics for P2P systems, a procedure to answer queries posed to the P2P system. Many other papers were identified with the similar claims including one from N. Arthorne, B. Esfandiari (2006), which proposed a framework building a 'bridge' to connect the peers.

Integrated Views

Many theories were found proposing to use 'views' to answer queries and establish data integration. One such survey "Answering queries using views: A survey" was done by Alon Y. Halevy (2001). It was argued using views to answer queries was a key technique to allow database systems to be maintained independently. Most theories for data view approaches are to define and materialize views over the databases, and use these views to answer queries. Views have long been embedded with database management systems for provide easy access to data from multiple tables. All major DBMS product like Oracle and Microsoft SQL server have the features of views. Views in these DBMS allow users to query and maintain data without direct contact and knowledge of the underlying tables. In term of data integration, using views can be an effective way of building data integration among different sources, and allowing data integration algorithms to be developed. In many cases, users submit queries over the views; the data integration system needs to reformulate the queries to refer to the data sources; the rewritten queries are then

optimised and executed on the source data. The majority of the theories proposing view are focused on optimising the process of convert queries for the sources and achieving high performance for process queries on global data sources, as the main issue of using view was regarded to be performance degradation. Queries on the views must be translated into sub-queries against the underlying tables. If the view itself is made up of complex, multi-table or views, even simple queries against the view can be resulted to complicated joins and take a long time to run.

Mediated Schema

Many theories were found in merging existing database schema and creating global database schemas for the purpose of data integration and running queries on them. A global schema provides a definition of all the available data and their relationships among all the existing database sources. It is the base of writing queries and getting target information from all underlying data sources. One such theory was proposed by Rachel Pottinger, Philip A. Bernstein (2008) in a recent paper, the authors claimed that a process similar to the traditional normalization for relational database can be applied for creating such global schema. Their algorithm not only generates the global (mediated) schema, but also generates view definitions, i.e., source-to-mediated schema mappings. They proposed a data integration framework to allow users to query multiple data sources based on a unified, mediated schema. Each query over the mediated schema is then translated into queries over the source schemas. The results of these queries are combined and returned to the user. In order for such scenario to happen, the creation of a global (mediated) schema becomes necessary. The process of creating such global schema can be like establish a relational database schema based on existing data elements (entities), but in this case, these are existing elements from different data sources.

There was another mediated schema theory from Maurizio Lenzerini (2002) who tried to use database schemas and the merge of them to create a new integrated global schema with mappings. In his theory, a data integration system is defined as a triple $\langle G, S, M \rangle$ where G is the global (or mediated) schema, S is the heterogeneous set of source schemas, and M is the mapping that maps queries between the source and the global schemas. Both G and S are relations, while mapping M consists of assertions between queries over G and S . When users pose queries over the data integration

system, they pose queries over the global schema G. The foundation of this theory includes the following concepts: A database over a schema is defined to be a set of relations. The database corresponding to the source schema S would be the set of sets of tuples for each of the data sources and is called the source database. Source databases can include disconnected databases. The database corresponding to the schema G is called the global database. The global database is built on mapping M with respect to the source database. The mapping M is crucial and it links together the G and S.

Schema Matching

In a survey paper by Erhard Rahm, Philip A. Bernstein (2001), they have summarized some of the theories on automatic schema matching, for which the primary purpose is data integration. In their paper, they present a taxonomy covering many of these existing approaches. A following diagram from the paper summarizes their claims.

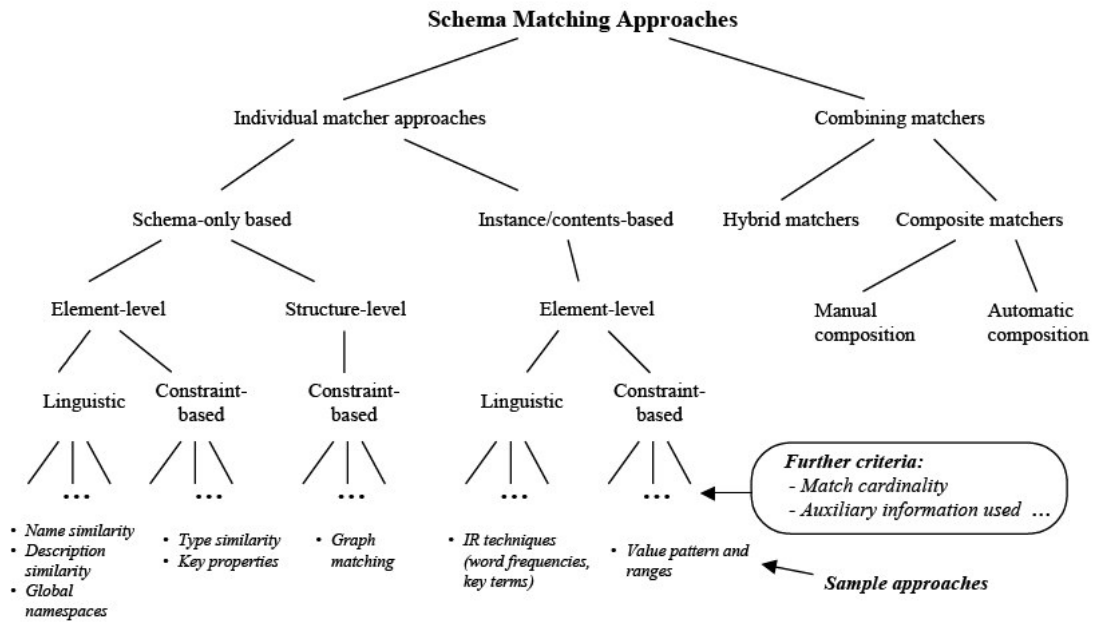


Figure 2. 1 Diagram by E. Rahm, P.A. Bernstein: A survey of approaches to automatic schema matching, 2001

In summary, schema matching can be done in different directions and into different level of detail. As illustrated in the above diagram, schema matching approaches can be categorized first into individual matcher or combining matchers, and categorization

using different aspects of the schemas are applied repeatedly. The paper also listed a set of sample approaches from the literature. They are categorized as following:

Prototype schema matchers (They are published prototype implementations fit for the classification criteria)

- SemInt (Northwestern Univ.)
- LSD (Univ. of Washington)
- SKAT (Stanford Univ.)
- TransScm (Tel Aviv Univ.)
- DIKE (Univ. of Reggio Calabria, Univ. of Calabria)
- ARTEMIS (Univ. of Milano, Univ. of Brescia) & MOMIS(Univ. of Modena and Reggio Emilia)
- Cupid (Microsoft Research)

Related prototypes (they offer functionality that is related to the schema matching approaches)

- Clio (IBM Almaden and Univ. of Toronto)
- Similarity flooding (Stanford Univ. and Univ. of Leipzig)
- Delta (MITRE)
- Tess (Univ. of Massachusetts, Amherst)
- Tree matching (NYU)

Summary

In summary, most of the data integration theories were focused on generating data query algorithms to enable and improve queries across different data sources. As data queries are always based on data base schemas, linking of different database schemas together could be the key for data integration. The review concluded that the key of data integration is a global (mediated) data schema, and the creation of such schema depends on schema matching and element mapping among different data sources. Because approaches including P2P integration, integrated view and schema matching would really work for data source from the same platform, if the databases are of different platform or even different formats including relational, object and XML databases, the global (mediated) data schema would be the only effective way for data integration.

Finally, data integration remains to be one of the most challenging tasks in the academic and commercial world, as it was argued by Patrick Ziegler and Klaus R. Dittrich (2004) in a paper called “Three Decades of Data Integration – All Problem Solved”. The author argued, even though data integration is one of the older research topics, there seemed no silver bullet solution and there would be none to be expected. The reasons would come from many directions, including there would be more and more business requirements for such integrations but may be not necessary funding coming with it; there would be data quality and security issues; and last, data integration infrastructure would be difficult to set up and maintain.

2.2.2.3 Functional Integration

Overview

Functional integration is the process of linking together of the component subsystems and ensuring that the subsystems function together as one system, utilizing and integrating functions provided from the subsystems. As it was argued by Thomas Wendt, Birgit Brigl, Alfred Winter (2005), functional integration happened when functionality from several application components were combined and used by all users or other applications. Although lots attentions are given to technologies for linking and integrating physically such as network protocols, we should make a distinction between physical and functional integration. In the following sections, we only discuss the functional part of system integration.

Integration Objectives

A functional integration is an aggregation of functions from subsystems to deliver the over-arching functionality. Integration involves joining the subsystems together through interfaces; those interfaces can be user interfaces which end users will interact or machine interfaces that sometimes are called application-programming interfaces (API). Functional integration is about determining the existing interfaces of the subsystems and the required interaction between them, so a new function can be created and new objective can be achieved. If the interfaces of subsystems do not directly interlock, the integration process has to provide kind of “glue” between them, which can be some kind of mappings in between functions and translations of input/output data.

Functional integration is about value-adding to the system, adding new capabilities that are only possible because of interactions and collaborations between subsystems. As it was argued by Patrick Ziegler and Klaus R. Dittrich (2004), in general, system integrations aimed at combining selected systems so that they form a new system and give users the functions as if they were from one single information system. There would be two categories of objective for integrations. First, integrations can be created over a set of existing information systems for providing a single information access point. Second, integrations can combine data and functions from underlying systems for providing more comprehensive basis for business needs.

Integration Foundation

Underlying computer networks, computers and their operating systems are the foundation of any integration and where it can take place. Functional integrations can be in the local machine, local area network and wide area network.

In a local machine, a number of software applications can interact with each other, which would result improved productivity and new functions for the end user. A typical example of this is the Microsoft operating system and the MS Office suite products. These software products and the integration of them are so powerful; they have dominated the PC market for a long time.

As most business and organization have their own local area networks (LAN), most information systems run in a LAN environment, so do functional integrations. Functional integrations on a LAN are the most common and very crucial in the IT industry. One of these examples is the company intranet site (Internet Information Service application). It not only can act as the central portal for many other applications, but the intranet site itself will most likely to be implemented with functional integration with other company applications, such as the network control systems and HR payroll systems.

Functional integration over wide area network (WAN) has a history almost as long as computer networking. A lot of standards and techniques for these integrations have been developed. As it was argued by B. Medjahed, B. Benatallah, A. Bouguettaya, A. H. H. Ngu, A. K. Elmagarmid (2003), Business-to-Business (B2B) technologies were

well before the current Internet age. B2B applications were among the first to take advantage of advances in computer networking. The Electronic Data Interchange (EDI) business standard is one of those early examples. Internet has made it possible for the masses of businesses to automate their B2B interactions. However, several issues have arisen, including scale, content exchange, autonomy, heterogeneity, and others.

In summary, local area network and wide area network provide the host for most of the functional integrations. Functional integration across organization network and even the Internet provides lot more powerful functions than those from a single machine. Internet and the TCP/IP protocol is the centre of these integrations in many cases, as connecting to the Internet is becoming a standard function for any organization network; development tools are becoming more advanced and implementation of such integrations are becoming more common.

Integration Architecture

According to Microsoft Patterns & Practices, Prescriptive Architecture site, functional integrations can use different patterns in terms of their integration methods. They are

- *Distributed Object Integration*
In distributed object integration, objects inside one application interact with objects in another remote application in the same way that they would interact locally with another object. Most of these integrations happen in the local machine, but they can also be implemented over networks.
- *Message-Oriented Middleware Integration*
In message-oriented middleware integration, the connected systems communicate by using messages that contain small packets of data. The main advantage is the communication can be asynchronous, which makes the messages less likely being lost during network or system failure and hence improve the reliability of the integration.
- *Service-Oriented Integration (through XML-based Web services)*
Service-Oriented Integration connects systems by enabling them to consume and provide XML-based Web services. The main advantage of this pattern is it can be easily implemented over the Internet and make the functional integration across different system platforms.

There are many framework approach theories of functional integration. One of them was from Markus Stumptner, Michael Schrefl, Georg Grossmann (2004), they argued that integration of autonomous object-oriented systems required the integration of object structure and object behaviour. In their paper, they proposed a set of tool, a meta-class architecture and a merge process of system objects. The integration would automatically take place according to the settings and rules that were decided and set up by end users. Their framework included components of Object Behaviour Diagram (OBD), UML activity diagrams, and web service coordination languages. They worked on the base of merged object type and consistency of their behaviours.

Summary

Functional integration is an important part in the system integration arena. The challenges come with it are from many direction as described above. As the world is becoming more and more connected, so are the information systems. More systems are designed to connect together with others including those from the past, being currently used and in future.

2.2.2.4 Workflow Integration

Overview

Workflow integration is automation of a business process, during which documents, information or tasks are passed from one participating system to another for action, according to a set of procedural rules. Workflow integration can be regarded as a special kind of functional integration or business process management system with focus on passing relevant data through many steps, applications and computers; during the process, a set of rules are applied for controlling the process and achieving the end result. The main benefits of workflow are automatic coordination and collaboration among participating systems and business.

Most of the workflow integrations take place across the organization-wide local area network and the Internet among different organizations' LANs. Typical workflow examples include automatic procurement process, web site publishing process and electronic time sheet processing. With availability of Internet, new workflow applications have developed, such as electronic auctions (e.g., Ebay), all-in-one travel

planning (e.g., Expedia), and automation of real estate purchase (e.g., Realtor). These new workflows are based on new e-services available on Internet. These new workflow systems are creating a lot of new business opportunities, as they not only streamline and speed up the business processes, but also reduce cost by choosing the cost effective services available on the Internet.

There are special challenges for workflow systems, as it was argued in a paper by German Shegalov, Michael Gillmann, Gerhard Weikum (2001). For those traditional and mostly internal workflow processes, there would be issues of integration of business objects, especially if they were from different platform. For new workflows on the Internet, there are issues setting up configurations and contracts with those e-services, so they could all work together.

Workflow Products

Since workflow processes have attracted high interest especially from commercial world, there is a large range of software products and providers. Many workflow integrations are used in specific business environments, there are products from various software companies, which can be adapted and customized in different client environment.

For all major software companies, the review had identified the following products:

Provider	Product Name	Comments
SAP	SAP	SAP is a suite of all kind of business applications; workflow is supported as part of the business process.
Microsoft	BizTalk	BizTalk server provides all the necessary backbones for building all kinds of workflow process applications.
IBM	WebSphere MQ Workflow	It is set of components based on IBM MQ series and WebSphere business Integration Server.
Oracle	Oracle Business Process Management	It is a set of middleware components for modeling, executing, managing, and optimizing business process applications.

There are numerous other workflow products, the following is just a small sub-set summarized and listed in the "Open Directory Project" site (retrieved June 17, 2008, from <http://www.dmoz.org/Computers/Software/Workflow/Products>)

- **Active Endpoints** - *Active Endpoints provides services-oriented integration solutions, and a BPEL-compliant workflow platform.*
- **BizFlow** - *Enterprise-level workflow and business process management platform, focus on web-based process delivery.*
- **Brightwater Software** - *The FlowRunner suite offers workflow, process management and improvement solutions. HTML, PDF, InfoPath form technologies are supported.*
- **Compuware UnifaceFlow BPM** - *UnifaceFlow allows business analysts to model business processes, and integrate them with existing applications, automatically test and deploy the process while providing tools to execute, manage and track process activity.*
- **FloSuite BPM Software** - *Business Process Management (BPM) solutions built on Microsoft.NET for legal and professional services organisations.*
- **Lotus Domino Workflow** - *Domino Workflow is a standalone product that works on top of Domino to provide our customers with the ability to develop, manage, and monitor all their business processes and help them eliminate the downfalls of paper-based work.*

Summary

A workflow would be regarded as abstraction of a business process. It consists of two or more sub-processes that would take place in a certain order. Such scenarios would likely happen in most business processes, so majority of attentions and effort for workflow integration have been come from and gone to the business world. The large number of products identified by the review was a proof. In academic world, workflow has also attracted a lot of attentions. A lot of theories and frameworks have been published in this area, which are discussed in the following solutions section.

2.2.3 Integration Solutions

The review has identified numerous proposals and solutions for all sorts of integrations, as they cover such a large area, and there are so many different challenges in various situations. New technologies and new business requirements are arising all

the time, they would push the development of integration solution even further. The followings are summary of theories that have attracted significant attention and would provide promising perspectives.

2.2.3.1 Online Database Integration

Database integration is the key for many system integrations. They can be implemented on a static base such as by importing all relevant databases into the central database, or on an online dynamic base such as by linking and querying underlying database whenever it is necessary. The later one seems to be more preferred and more promising in the ever growing and changing IT world. Importing databases to a central one and maintaining it would be an impossible task with the ever growing and changing world, even with the latest system and hardware provided. Therefore, the review has found the majority of database integration research and development effort are focused on building unified views and global schemas so end users can execute their queries and extract data without knowing the underlying database. Numerous techniques and papers have been dedicated on this area. They are ranged from data extraction from existing database, underlying database schema merge, mediated schema creation and view creation.

There are many proposals for frameworks of implementing systems that allow users to create and use data integration system on top of multiple data sources. One of such example is “BioXBase” system proposed by Pavithra G. Naidu, Mathew J. Palakal, Shielly Hartanto (2007), it is a framework for creating a user centric query system to extract useful biological information from the Internet through providing unified views. It was argued that the BioXBase system would be used to develop a conceptual framework for discovering self-describing web, and the framework would be dynamically restructured. The authors claimed their results indicated that knowledge discovered from semantically distributed heterogeneous web sources had produced more significant data than those from a local database system.

There are also methods for matching attributes of data elements on the fly for data integration. One of them was proposed by Cecil Eng H. Chua, Roger H. L. Chiang, Ee-Peng Lim (2003). They proposed methods for matching attributes with different data type, scales, abstractions, erroneous or null values, and with synonyms and homonyms.

It was argued, when schema information is either inadequate or inappropriate, the best way for data integration was to use instance-based attribute identification and matching.

There are many theories for improving performance of online database integration too. One of these was proposed by Janusz R. Getta (2006). They proposed an optimisation mechanism for online data integration in a multi-database database system. The mechanism was to continuously append the increments of data and re-computes a data integration expression after each appending process. They also proposed a new system of operations for processing the increments of data sets against already transmitted data for improving the efficiency of the appending process. Finally, they compared three types of optimisation techniques, which included reduction of input data, elimination of materializations and reduction of arguments. In summary, it is a data integration framework system that keeps the database (data set) optimised in an ever-changing environment.

2.2.3.2 Meta Data Approach

It was argued by Goksel Aslan, Dennis McLeod (1999), mediation of metadata and ontology across database boundaries are key aspects of interoperation among data-intensive systems. Data and metadata need to be mediated across multiple databases for information sharing and exchange. Metadata is data about data; it can exist in many forms and in many places. For the purpose of data integration, it means descriptions of the database, entities and attributes. In data integration, metadata are mostly required for all source databases; based on them, a global database, schema and views can then be defined. The results are mediation between the end users and the data sources. The theory proposed was to transfer remote metadata (conceptual schema) into the local metadata, and use them as a common platform for information sharing and exchanging. There are many other ways to create the mediated metadata and to improve its quality for purpose of data integration on it. The review has found most of them can be categorized as followings:

System Frameworks

As metadata can act as critical component for discovering relevance of certain data sources, lots data integration frameworks have been built for specific working environment and data area. For example, a methodology for semantic integration of

metadata in bioinformatics data sources was developed by Lei Li, Roop G. Singh, Guangzhi Zheng, Art Vandenberg, Vijay Vaishnavi, Sham avathe (2005). It was aimed for effective and efficient use of the fast growing heterogeneous bio-informatics data sources, by monitoring, clustering, and visualizing bio-informatics metadata across different data sources.

Generic Algorithms

Because interoperability among databases has become so important among DBMS and application software, even within a single organization, data from various sources must be integrated for helping to achieve organizational goals. Generic metadata methods and algorithms have been developed to fulfil such needs. One such example is a relational algebra for data/metadata integration in a federated database system that was developed by Catharine Wyss, Dirk Van Gucht (2001). In their paper, a relational algebra called LOGSPACE was proposed, the main objective is to enable the integration to produce a relational federated system, so existing techniques for query evaluation and optimisation can still be applied. Later on in 2005, a relational language for metadata integration was developed by Catharine M. Wyss, Edward L. Robertson (2005). It was called Federated Interoperable Relational Algebra (FIRA). FIRA has many properties such as compositionality, closure, a deterministic semantics, a modest complexity, and support for nested queries. A declarative query language called Federated Interoperable Structured Query Language (FISQL) was also proposed.

In summary, metadata can provide the foundation for data integration in many cases, especially when the creation of global database schema is necessary, so does it for data merger and schema matching. As it was argued by Andreas Tolk Charles D. Turnitsa (2007), if the systems interfaces and data exchange requirements would be captured in metadata, the metadata would then be used identify exchangeable data. Although the creation and maintenance of such metadata are still one of the most challenging tasks for researchers and practitioners, purpose built Metadata can allow systems to exchange information based on self-organizing principles. They can make data integration much more efficient without user or developer intervention.

2.2.3.3 Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) is software architecture for solutions where functionality is grouped and packaged as services, overall system functions or business

processes can be achieved by linking and interoperating with those services. It is particularly popular in government IT departments. In a paper by Andréa Matsunaga, Maurício Tsugawa and José A. B. Fortes (2007), it was argued, an SOA was for a set of government processes, which would consist of a collection of loosely coupled services. The service would interoperate with each other. Interoperability is achieved by using standard languages for the description of service interfaces and the communications among services. They then went on to propose a Command-Line Application Wrapper Service (CLAWS) framework base on web service and the SOA architecture.

SOA is becoming more and more popular in system integration even in implementation of single systems. It has lots of advantage over those old technologies; such as the rigid point-to-point electronic data interchange (EDI). The advantages of SOA can be summarized as followings:

- It inherits all benefits of modular software development, but instead grouping them by services.
- It allows of platform independent system development and integration, as long as the services provides commonly understood interfaces.
- As it is loosely couple in its design and implementation, it allows more rooms for individual service to be adapted to changing technologies, accommodating legacy application. As it was argued in a paper by Andréa Matsunaga, Maurício Tsugawa and José A. B. Fortes (2007), the main benefits of SOA and WS were from the decoupling of service interfaces from their implementations. The principle of decoupling is to allow a service to be created, and then to be updated, replaced or modified without changing its interface and affecting other services.”
- The common service interface allows for quick and easy creation of business process from existing services. It also allow for easy maintenance for these business process.
- With implementation of web service, it allows software development and business to connect to services available on the Internet anywhere, which is virtually unlimited.

SOA can be built on many different platforms and with many different development techniques, but the one technique which has stood out to be used most for

implementing SOA is Web Service. As it was argued by Andréa Matsunaga, Maurício Tsugawa and José A. B. Fortes (2007), Web Service (WS) was a widely accepted technology for implementing SOAs, WS would use WSDL (Web Services Definition Language) and SOAP (Simple Object Access Protocol) as the standard languages for definition of, and communication among, services, respectively. Web service includes a set of specification and rules to control communication and interaction among systems and computers over a network. The W3C group maintains a set of these standards, but it is up to the individual software provider to implement these into their product. There are many different favour of implementation from different software companies, such as Microsoft and IBM. Web service was invented to make use of Internet and overcome difficulties arising from different system platforms. Since they are based on 'open' standards and heavily focused on Internet, web service techniques have been adopted by more and more software providers and system development, especially for building SOA based components and e-services.

In general, SOA is adopted for functional integration and workflow integration because of the Web Service technologies used of implementation has adopted many standards for interface design and communication, especially for service discovery, workflow orchestration, reliable messaging, security, etc. SOA is more important in large organizations, such as government agencies than it is for small business. As it was argued by Andréa Matsunaga, Maurício Tsugawa and José A. B. Fortes (2007), the main driving force for adoption of SOAs for digital government were from the fact that, once the services were implemented and known, government processes would be easily created, maintained, integrated and reused. This, in turn, can enable agile government processes and facilitate new forms of cross-agency interactions and public access to government functions and information. It was quoted in the paper that the USA government's Federal Enterprise Architecture has specified SOA's and WS technologies to be used at all services layers.

There many theories for extending the application SOA and web service technique. One of these was proposed by Bastin Tony Roy Savarimuthu, Maryam Purvis, Martin Purvis, Stephen Cranefield (2005). They proposed that adaptable web service to be implemented for acting as agent of business workflow management system. It was argued that the rapidly changing business environment had demanded for flexible and

adaptive workflow systems. The advantages of such approach are Web service can be easily integrated into a workflow system, and the web service itself as an agent can also easily connect to any other web service dynamically.

2.2.3.4 B2B E-commerce Frameworks

For functional integration, one of the main driver and also the main beneficiary is B2B E-commerce. E-commerce can mean a lot of things, including any online stores and anything involved running a business using a computer. B2B E-Commerce is focused on how business, specially their computer systems, can be integrated with each other for completing business transactions. B2B E-commerce frameworks have long been the solutions for lots functional integration and workflow integration. As it was argued by B. Medjahed, B. Benatallah, A. Bouguettaya, A. H. H. Ngu, A. K. Elmagarmid (2003), B2B E-commerce had been around for almost three decades, many businesses especially the banking industry had used secure frameworks for sharing and exchanging data. In the early day, Electronic Data Interchange (EDI) standard was the most widely used and earliest framework. Later, as software development and networking technology has advanced, new methods and frameworks became available. They include component-based frameworks for distributed messaging and computing that can securely run on public computer networks. AS more and more business rely on connecting to their business partners and Internet, inter-enterprise frameworks are becoming the 'silver bullets' for tomorrow's virtual enterprises software applications. There are also special challenges for B2B frameworks even with all foundation technologies readily available. A successful B2B framework must include solutions for process-based integration of services, dependable integration of services, and support of standardized interactions, security, and privacy.

The review has identified numerous B2B E-commerce framework software solutions. Some of them are summarized in the followings:

ebXML

It is an XML-based e-commerce initiative and its name stands for electronic business XML. ebXML was started in November 1999 by UN/CEFACT and OASIS.

UN/CEFACT is the United Nation's Centre for Trade Facilitation and Electronic Business (UN/CEFACT). OASIS stands for Organization for the Advancement of Structured Information Standards. Both organizations are behind the ebXML initiative

for providing an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties. CNET UK has selected ebXML as finalist for the "Most Promising Technology of the Year for 2003" Award. In summary, ebXML is a set of standards for business to store their data and do communication in XML. By using ebXML, companies could exchange business messages, conduct trading relationships, communicate data and define and register business processes.

WS-BPEL

It stands for Web Service Business Process Execution Language. WS-BPEL is a language for the specification of Executable and Abstract business processes. It has recently been proposed for formal specification of business processes and interaction protocols for Web services. The latest version 2.0 was published by OASIS in April 2007. WSBPEL defines an interoperable integration model that facilitates expansion of automated process integration in both intra- and inter-corporate environments. WS-BPEL extends the Web Services to support business transactions. It defines a new interoperable model that could be used for the expansion of automated process integration in both the inside and outside the enterprise network.

B2BOOM

It stands for B2B Ontology-Oriented Middleware, proposed by Ejub Kajan, Leonid Stoimenov (2005). It is a framework for building and applying semantic interoperability to enable B2B e-commerce. It is based on infrastructure of Internet and web service. It proposed to use Ontology Web Language for Web Services (OWL-S), to describe what a service does, how it works, and how to map the process model onto detailed specifications by using the OWL sub-ontology: the profile, the process model, and the grounding. They argued that semantic interoperability was an essential attribute for bridging the complexity of heterogeneous environments and may allow flexible mediation between business partners on a conceptual basis. With their emerging approach, they could transform the Internet and Web from a static collection of data into a distributed multiprocessor machine making the content machine readable and machine process-able. They argued B2BOOM could produce full semantic interoperability and bring B2B to its full potential.

Others

In 2000, a summary of B2B framework at the time was done by Shim, S.S.Y., Pendyala, V.S.; Sundaram, M., Gao, J.Z. (2000). It listed five frameworks (eCo, BizTalk, OBI, cXML, RosettaNet) of the time and compared them in features including industry target, security, communication protocol, repositories, message format, query mechanism, scalability and ontology.

Table 1. Comparison of B2B frameworks (MSMQ: Microsoft Message Queue; SSL: secure socket layer).

Feature	eCo	BizTalk	OBI	cXML	RosettaNet
Industry target	Unspecified	Unspecified	MRO materials, nonproduction supplies	MRO, office supplies, books, and so on	IT and electronic components
Security	Optional	Leverages existing standards	SSL with HTTP; digital certificates and signatures	Authentication in message header	SSL with HTTP; digital certificates and signatures
Communication protocol	HTTP	Wide variety: HTTP/MSMQ	HTTP	HTTP, URL form encoding	HTTP/CGI
Service discovery	Extensive support	Supported	Not addressed	Not addressed	Not addressed
Repositories	Locally maintained	Centralized repository	Owner's responsibility	Not addressed	Not defined within scope of standard
Message format	XML documents	BizTalk documents based on Biztags	Encapsulated EDI documents	XML documents	Valid XML documents
Query mechanism	URL-based	Not addressed	Not applicable	Not addressed	Not addressed
Scalability	Sufficient	Centralized repositories and processing may limit	One-to-one framework does not impact scalability	Scalable, based on XML DTDs	Allows extension of implementation guidelines
Ontology	Common Business Library	Collection of Biztags	EDI X12 data dictionary	Collection of XML tags	Technical and business dictionaries

Table 2. 1 Summary table from “Business-to-business e-commerce frameworks” by Shim, S.S.Y., Pendyala, V.S.; Sundaram, M., Gao, J.Z. (2000)

Summary

In summary, Internet and the use of it is the foundation for most B2B E-commerce. As Internet has become the global information super-highway, most business have connected to it and tried to make the most out of it in the same time even by adopting new process and practice in order to achieve that. It was argued by B. Medjahed, B. Benatallah, A. Bouguettaya, A. H. H. Ngu, A. K. Elmagarmid (2003), interoperability would be most important in B2B applications because there were so much difference in the way businesses would operate, the systems they would have. Most difficulties were caused by systems' autonomy and heterogeneity. It has also become clear that Web Service and XML are the central elements for these frameworks, whether they are the communication mechanism or format for data. It can be concluded that with the powerful driving force from the business world, more B2B framework will be developed and put into practice. The challenges and solutions for functional integration

among business systems will remain as one of the hottest area for future research and development.

2.2.4 Summary of Application Integration

The review has gone through literatures in a wider area of system integration. It was found to be a big topic attracting a larger amount of research and commercial interest. Most of them were driven by new technologies, including Intranet, XML and Web Service. The review had identified a large amount of integration frameworks as well as commercial products. They were for various purposes, from data integration to functional integration. A lot of the research effort had been focused on creating new frameworks or methods for selected integration purposes. However, they were not often supported by sufficient evidence of general cases or backed by successful commercial products. The review has found very few papers discussing general integration development and research method.

2.3 Enterprise Application Integration

2.3.1 Overview

The need for system integrations within an organization has led to the evolution of enterprise application integration (EAI). In its April 2001 report for AIIM International, "Enterprise Applications: Adoption of E-Business and Document Technologies, 2000-2001: Worldwide Industry Study", Gartner defines EAI as the unrestricted sharing of data and business processes among any connected applications and data sources in the enterprise. As argued by Nina Reiersgaard, Hilde Salvesen, Stig Nordheim, Tero Päivärinta (2005), EAI is an approach to integrating core business processes and data processing in the organization. EAI is often a process of integrating applications within a single organization together in order to simplify or automate business processes, at the same time it would maintain the existing applications or data structures. The review had found EAI would be regarded as sub-set of system integrations, which take place within the boundary of the corporate enterprise.

EAI is considered very important because of its potential effect on the enterprise system. As argued by Gunjan Samtani and Dimple Sadhwani(2001), the origin of EAI would be from the need for providing a full solution to share and exchange data

between ERP, CRM, SCM, databases, data warehouses, and other important internal systems within the company. The review has identified numerous papers in literature, EAI products and technologies in the commercial world. As argued by Piyush Maheshwari (2003), the essential value of EAI would be from many directions, such as automation brought into the system, better efficiency and less maintenance for the enterprise system. In summary, the benefits of EAI would be described as followings:

- 1) It would allow users to interact multiple systems from one central interface.
- 2) It would allow users to access or combine data from multiple systems.
- 3) It would allow users to access or combine functions from multiple systems.
- 4) It would allow users to combine and automate processes from multiple systems.

There would be many challenges for EAI. One of the main ones would be that the various systems to be integrated would run on different operating systems, use different database platforms and different development tools. Any large organization would have systems supplied by different vendors who would provide different level of support or no support at all. In summary, because of the variety of business systems and the various integration requirements, an EAI would even face more challenges than system integration across multiple organizations.

2.3.2 Challenges & Solutions

2.3.2.1 The Evolution

In a paper by Jinyoul Lee, Keng Siau, and Soongoo Hong (2003), they had summarized the evolution of information technology related to system integration in the following diagram:

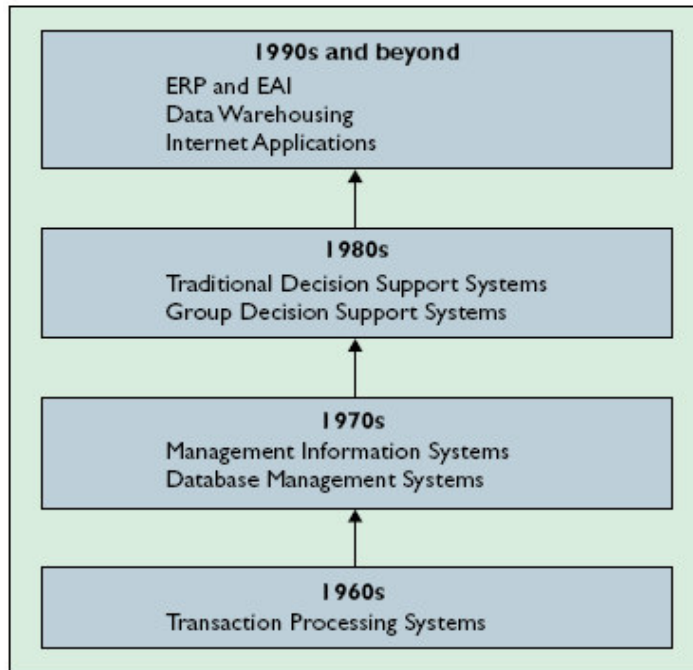


Figure 2. 2 Diagram by Jinyoul Lee, Keng Siau, and Soongoo Hong (2003)

They argued traditionally, information systems were developed for specific functional requirements, which would include database, user interface and business process. As organizations become more complex and diverse, it becomes nearly impossible for organizations to carry on with their business without enterprise integration. The EAI concept emerged in the mid-1990s aiming for achieving integration with lower costs and less programming. They further described the difference between traditional integration and the EAI in the following diagrams:

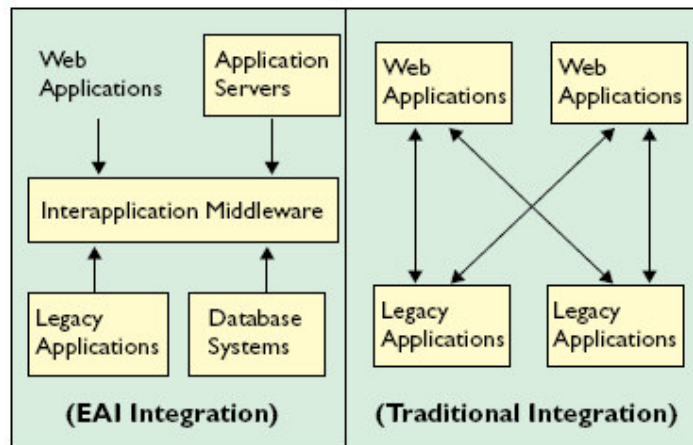


Figure 2. 3 Diagram by Jinyoul Lee, Keng Siau, and Soongoo Hong (2003)

They argued that traditional integration would involve a lot of development and coding, they would also create specific and sometimes multiple integrations within the

organization. On the other hand, EAI uses special middleware that would connect to multiple systems for the integrations. All applications would communicate with each other through a common interface. Thus, EAI would avoid extensive programming and reduce cost.

Many new challenges had been raised for EAI, including the following:

1) Technical integration vs. Behaviour integration

Technical integration including software and hardware is only one aspect of integration. The biggest challenge would come from the behavioural integration. Change of the organization from the integrations would cause difficult and sensitive issues, if they were not managed properly, the result would be a disaster. The integration would only succeed if the organization welcome and adapt to the new enterprise system. To achieve the goals of EAI, both technical and behavioural integration must succeed.

2) ERP approach vs. EAI approach

ERP has been a dominant organizational trend since the early 1990s. As argued by Bingi, P., Sharma, M.K., and Godla, J.K. (1999), analysts stated that 70% of Fortune 1,000 firms currently have or will soon install ERP systems and the ERP market is predicted to expand into the foreseeable future. The concept of enterprise integration was pioneered with ERP, which is an enterprise-wide software solution for different functions in an organization. ERP is internal focused by providing all the functions and processes inside the package. It is often very expensive and would require changes of business process and even reengineering. It does not focus or cope well with the changes of external environment. On the other hand, the basic concept of EAI is integration with low costs and little coding using existing applications. The principle of EAI is that it allows a business-oriented approach rather than standard ERP business process reengineering. EAI's benefits would include fast delivery low development cost.

The ERP Factor

It will not be possible to discuss EAI without talking about ERP, as there were numerous ERP products that offer EAI solutions, or do so as at least claimed by their vendors. ERP market is huge and has big influence, according to Gartner's report by

Alex Soejarto, Michael von Uechtritz, Ben Pring (2009), all of the top ten vendors of ERP product and service in North America have revenues exceeding one billion dollars. In a paper by Marinos Themistocleous, Zahir Irani, Robert M. O'Keefe and Ray Paul(2001), ERP benefits would include:

- 1) They offer solutions to the problems of legacy systems
As ERP would replace them.
- 2) They reduce development risk
Most ERP package would offer a complete set of functions for a business, thus make any internal development unnecessary.
- 3) They would increase global competitiveness business efficiency
With implementation of ERP package, the standard processes would be implemented or forced onto the organization.

However, the drawbacks of ERP packages had also been identified:

- 1) Implementation complexity
ERP systems would likely to be complex and included numerous modules.
- 2) Integration problems
The internal structure of ERP would be unknown and special interface for integrations would always needed for any existing business systems.
- 3) Customisation problems
ERP package would not cover all the business needs and most business would have and want to maintain their unique characteristics.
- 4) Over budget and late projects
Most ERP are large packages and would involve high cost.
- 5) Organisational change and resistance to change
ERP implement would inevitably bring changes to the organization, and they would likely to be big ones.
- 6) Problems with business strategy and competitive advantage
The business has to learn and adapt to the processes from the ERP implementation.

It was quoted in the same paper that the 96.4% of ERP implementations fail whereas 70% of ERP implementations do not achieve their estimated benefits. The paper hence argued ERP would not offer integration solutions effectively; rather they bring up the

need for integration. If an organization wants incorporate other applications with ERP system, it would most likely encounter serious integration problems. However, the paper quoted most EDI applications were integrated successfully (81%) with ERP infrastructure because EDI technology follows similar concepts to application integration.

Similar views were also reflected in many other papers, including one from Gudivada, V.N. Nandigam, J. (2005), who argued the ERP implementation would be a way to replace existing systems for achieving enterprise integration, but due to its high cost (licensing, implementation, and maintenance costs), EAI had become a more attractive way for enterprise integration.

2.3.2.2 The Challenges

In a paper by Ian Gorton, Dave Thurman, Judi Thomson (2003), the challenges of EAI were summarized in the following areas:

1) Scale

Requirements for integration would come from the changes of business or systems. The changes of business and requirement would keep on going after the integration, especially for data integrations. The integration must be able to scale to handle large numbers of new data sources, which would likely to change in format or in quantity in a long run.

2) Dynamic configuration

The EAI technologies must use configuration to interact with an existing systems or data source, as and change to the EAI solution would be costly and time consuming. Ideally, it should be able to simply adapt and interact with new systems or data source with dynamic configuration.

3) Semantics

The interface of the integration must be reflected with semantic descriptions of how the interfaces should be used. This will enable true dynamic discovery and integration. As many modern programming technologies including Java and Web Services, support dynamic discovery of the syntax of a service or object interface.

4) Finding Relevant Data

For data integration solutions, especially for science application, the ability of

integration to handle huge numbers of data sources is critical. The integration must be able to rapidly discover and use new sources of relevant data, as the organizations cannot afford the expense or delay of to build and/or deploy complex integrations for new data sources.

Many other papers and article were also identified to have similar views on the topic with slight difference. One such article from the Microsoft argued the challenges of EAI would be from the following areas:

1) Application Semantics

Application semantics are different from application to application. For instance, how you define a customer within one application is very different from how you'd do it in another. So is it for business processes or functions. The EAI solution must include such transformation abilities to handle these semantic differences.

2) Information Content

Data integration is central for many EAI solutions. The solutions must be able to retrieve, transform and display the information content in real time to suit the target applications.

3) Platform Heterogeneity

EAI solutions must be able to handler operating system and interface differences, or platform heterogeneity. EAI solution must use the best suitable approach to interface with source or target systems according to their platforms. Many EAI problem domains consist of mainframe-based applications, ERP packaged, CRM applications, transaction processors, and various database platforms.

2.3.2.3 Solutions

The review has identified numerous products and solutions for EAI. Primarily they include enterprise resource planning (ERP), customer relationship management (CRM), supply chain management (SCM), and business intelligent (BI) products. For example, there was a large list of EAI solutions from the “EntirelyOpenSource” web site (retrieved June 10, 2009, from <http://www.entirelyopensource.com/oss-directory/eai-enterprise-application-integration>). However, the review was focused on theories and framework of the development for general EAI solutions. The review had identified and discussed

them in the following categories, which were proposed in a paper by Khubaib Ahmed Qureshi (2005).

EDI based integration

Electronic Data Interchange (EDI) provides data transformation in a compact and standard form (ANSI X12, UN/EDIFACT) using Value Added Networks (VAN). In a paper by Marinos Themistocleous, Zahir Irani, Robert M. O’Keefe and Ray Paul(2001), it was argued more company integrated EDI applications (e.g. procurement, ordering and invoicing) with their ERP systems than those with other systems. The concepts of EAI and EDI technology were similar, including extracting, translating, formatting and exchanging data between disparate applications. However, the Application Integration technology (AI) is in a more advanced way than those in EDI as illustrated in the following figures from the paper.

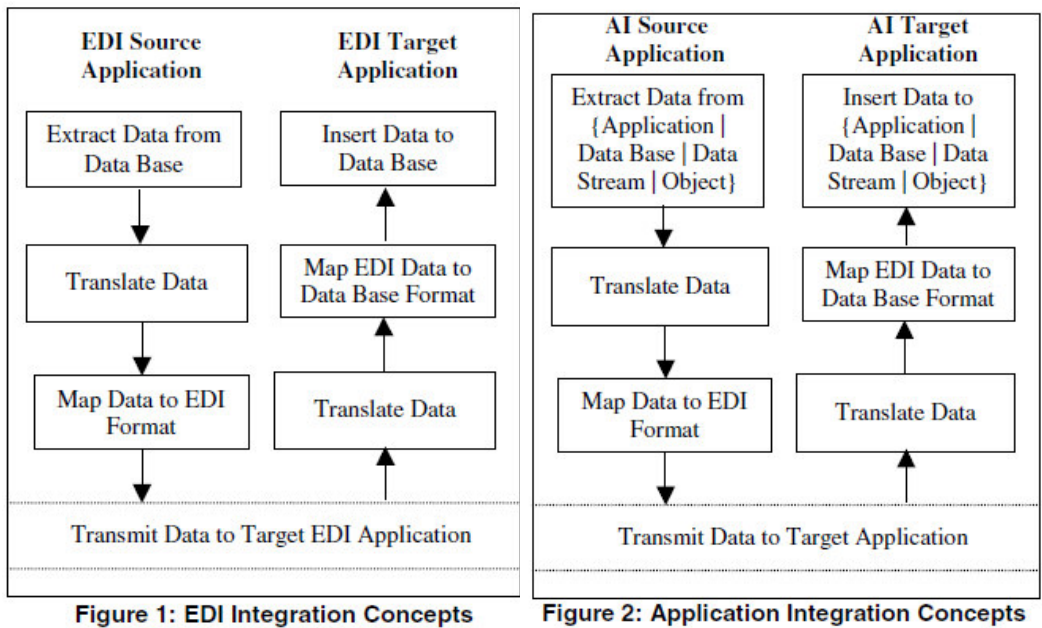


Figure 2. 4 Figures by Marinos Themistocleous, Zahir Irani, Robert M. O’Keefe and Ray Paul (2001)

Although EDI was listed as an EAI approach by Khubaib Ahmed Qureshi (2005), since the extra overhead of VANs, EDI approach is more often only used in B2B situation and being replace by Internet based technologies. The review had identified a few companies specialized in EDI over VANs, but none of them seemed to be related to EAI. No EAI framework was found associated with EDI, many other papers were

found arguing for using other means such as Web Service over the traditional EDI to implement integration solution. They would be discussed in later section.

Component middleware based integration

As it was also argued by Bichler M, Segev A, Zhao JL.(1998), component-based e-commerce technology was a trend towards resolving the e-commerce challenge at both system and application levels. As defined in a book by Sametinger, J. (1997), components are software with specific functions that can be accessed by other software through a contractually specified interface. They are self-contained, clearly identifiable and perform specific functions. Component middleware is a framework for creating components for integration solutions. The advantages of such approach would be:

- 1) Components can be defined and developed independently, which would make the integration development more efficient.
- 2) Existing functions would be included in integration components, which would speed up the development and reduce cost.

The disadvantages of component approach would be from its tight coupling nature, although many techniques have been used to lessen it including using message for communication, incorporating XML and database support in the framework. Another issue is its lack of capabilities dealing with presentation layer, which is often required as part of integration solution.

Many component based framework have been found in the review including one from Jin Yu, Boualem Benatallah, Regis Saint-Paul, Fabio Casati, Florian Daniel, Maristella Matera (2007). They proposed abstract model for automatic generation of presentation components for integration. They argued the framework would reduce the effort required for UI development by maximizing reuse. The components and composition are described by XML-based language. They also develop a proof-of concept system to prove the model can also easily used, even with different languages and/or component technologies.

In another paper by Piyush Maheshwari(2003), it was argued the technologies based on open standards such as CORBA, J2EE and .NET could provide a solid infrastructure for building a component-based framework. Advanced Java technologies, distributed object systems with CORBA and data integration with XML

would all be used for a component-based framework. The following are summary of the technologies:

1) Advanced Java technologies

Java 2 Enterprise Edition (J2EE) was a major initiative to provide a complete computing platform for Java programmers. J2EE builds on many components on the standard Java environment: Java Server Pages (JSP), Java Servlet, Java Naming and Directory (JNDI), Java Data Base Connectivity (JDBC), Java Messaging Service (JMS), Java E-mail, Java XML support, Java to CORBA mapping and the key component used in this paper, the Enterprise JavaBean (EJB)

2) Distributed object systems with CORBA

Common Object Request Broker Architecture (CORBA) is a conceptual “software bus” that allows applications to communicate with one another, regardless the platform and the language. CORBA is as middleware for distributed objects. The Object Request Broker (ORB) is a middleware component that implements the CORBA bus and acts as a broker between the client and the object. A Distributed Object System is a distributed system in which all entries are modelled as objects. This has become popular for combining the functionality of different applications, avoiding the risk in traditional application integration that typically involves low-level programming for the connection. CORBA is also regarded as good fit for object-oriented design with benefits of OO approach.

3) Others

Other integration methods include Remote Procedure Calls (RPCs), using an object-oriented database management system (OODBMS), Microsoft .NET. RPC is similar to COBRA, but RPCs are procedural and CORBA is of an object-orientated model. OODBMS is a very data-centric, it does not allow integration of non-database applications. .NET is similar to CORBA, but it does not have platform independence.

It was argued by Bichler M, Segev A, Zhao JL. (1998), Component-based programming with Microsoft ActiveX/COM or JavaSoft’s JavaBeans was popular programming paradigm of choice.

Workflow based integration

It was argued by Michael Stonebraker(2002), workflow systems were oriented toward procurement, and the focus was on process flow. Workflow technology is most important for automating business processes involving access to multiple applications. Workflow is often target for integrating, automating, and monitoring.

In a paper by Myungjae Kwa, Dongsoo Han, Jaeyong Shim (2002), they propose a framework supporting dynamic interoperation between heterogeneous workflow systems and workflow-based dynamic Enterprise Application Integration (EAI). It included four major components: The workflow engine, Adapter, Service Interface Repositories (SIRs), and XML messages. Workflow engine handles both internal and external sub-processes. The adapter transforms internal and external processes. SIRs contains the service information of other systems. XML messages enable the communications between workflow systems and enterprise applications.

There are two types workflow integration: Static and Dynamic.

1) Static interoperation.

In this case, potentially connectable workflow are predefined, workflow process execution follow the pre-defined path. It is not easy to add a process or change the configuration at run time.

2) Dynamic interoperation.

In this case, the binding of the sub processes is performed at run time according to binding rules. This form of interoperation has several advantages including that the designer needs not predefine all of the potentially connections, and the process definition is simpler. Moreover, the configurations of interfaces can be changed and new processes or systems can be added at run time.

In this framework, the workflow-based dynamic EAI is proposed based on the sub-flow task model and the multi-tiered dynamic state transition model. By encapsulating and dynamically binding sub-processes, it would increase scalability, flexibility, and interoperability of a workflow system.

The review has identified a large collection of workflow-based frameworks, including many open source workflow engines in Java, which would be found in "Open Source Software in Java" web site (retrieved June 10, 2009, from <http://java-source.net/open-source/workflow-engines>). Many software companies offer their own workflow integration

framework or product. The major software company Microsoft offers its .net Windows Workflow Foundation library included its development framework Visual Studio .Net suite, which is catered for full integration development.

XML based integration framework

XML would be used in EAI at several levels including data interchange, schema transformation, metadata management, process integration, and even message persistence. The main reason behind XML based approach is to enable communication between applications without mediation facilities. Business systems will be interconnected in terms of largely agreed upon XML documents. Those XML documents have common set of XML schema. The EAI framework server executes business process logic, passes requests in data flow, send or receive data with downstream applications. The content of XML document can be freely manipulated to meet the needs of the business requirements of the integration.

Application integration represents the larger problem of moving information between applications and data stores for any business purpose. XML provides a common mechanism for data exchange and integration, with a variety of applications supporting a variety of design patterns. What XML brings to the EAI is not great technology, XML and some of the derivative standards have been forcing the EAI vendor community into leveraging standard mechanisms, including XSLT and RosettaNet, within their now proprietary products. The use of XML within the enterprise allows easy migration to a strategic B2B information exchange platform. It would simply push the XML documents to B2B integration servers.

However, as pointed out by Khubaib Ahmed Qureshi (2005) the disadvantages of XML approach would be:

- 1) As the integration depends on the XML document standard (schema), business applications must be able to understand and adapt to those standards in their communication.
- 2) Because of the verbose nature of XML, exchanging data in XML would increase overhead of the integration.

In a paper by Vishnu S. Pendyala, Simon S.Y. Shim, Jerry Z. Gao (2003), they proposed an XML Based Framework containing the following components:

1) XML Interface

It is part of the enterprise applications for doing DBMS to XML conversion and vice versa.

2) Enterprise Integration Engine

It handles the core functionality including transforming XML data in one format to the other.

3) EAI portal

It is used to configure and define events and processes for different sub-systems.

A diagram of the framework extracted from the paper is displayed below:

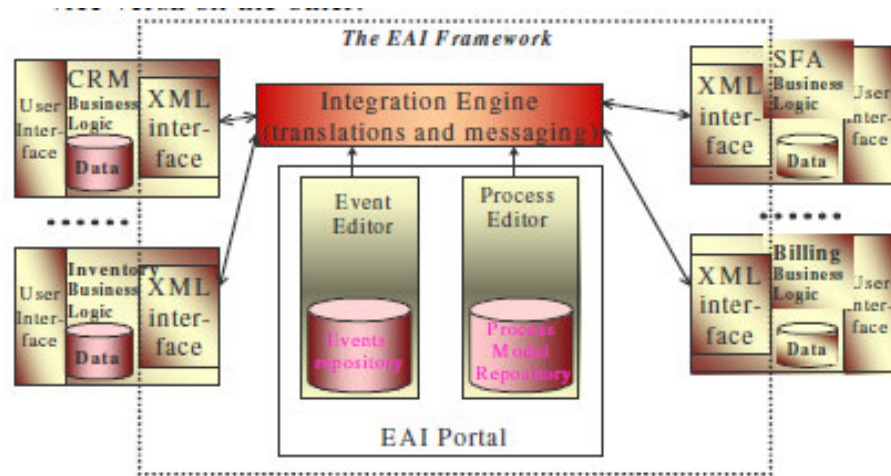


Figure 2. 5 Figure by Vishnu S. Pendyala, Simon S.Y. Shim, Jerry Z. Gao (2003)

The following are the advantages of the framework argued by the authors:

- 1) By using XML, the data and integration is more transparent. Compared to middleware approach which more like closed black boxes, XML based approach would increase the flexibility of the solution and also the extensibility and scalability.
- 2) The EAI portal is a single point control; it makes it easy to intervene by the administrator if it becomes necessary.
- 3) Since it uses XML schemas and XSLT style sheets to model the business processes, it is easy to query the business processes and modify them to suit changes in the environment.

- 4) Because it is based on XML schemas and style sheets, the architecture of the framework provides richer and more flexible data structures. The interactions are document-based, and it makes it easy for presentation of information.
- 5) The framework does not involve any programming for end-user to set up new integration.

Microsoft BizTalk server is also one of those most commonly used XML based integration framework. BizTalk server uses BizTags (standardized set of XML elements and attributes) for the documents passed around applications. BizTalk Document is a kind of SOAP message. The BizTalk framework contains three layers: Business application, BizTalk Framework Compliant Server (BFC), and Transport. Business applications interact to each other by sending business documents via BFC server. Special BizTags are used for ensuring reliable delivery of the documents.

RosettaNet is another XML based framework for data and process interchange with e-business defined by a consortium of product vendors and end users. According to the RosettaNet site (retrieved June 10, 2009, from <http://xml.coverpages.org/rosettaNet.html>), *“RosettaNet is a non-profit consortium of more than 400 of the world's leading Information Technology (IT), Electronic Components (EC), Semiconductor Manufacturing (SM) and Solution Provider (SP) companies working to create, implement and promote open e-business process standards”*. RosettaNet defines standard messaged data using XML, and standardized process flows to react to standard business events. What's significant about RosettaNet for the EAI problem domain is that it brings a nice process integration standard for use between, as well as within an enterprise, but currently it is more B2B oriented.

Many other XML based integration framework were identified including Concordia XML Framework, eCo, Commerce XML (cXML), Electronic Business XML (ebXML).

Web services based integration framework

As argued by Gudivada, V.N. Nandigam, J. (2005), *“a web service is an interoperable unit of application logic that transcends programming language, operating system, network communication protocol, and data representation*

dependencies and issues. It is an infrastructure for developing and deploying distributed applications.” Web service would be one of the most used technologies for integration. The benefits of using web service for EAI were summarised in a paper by Gunjan Samtani and Dimple Sadhwani (2001). They argued web services were:

1) Simple

With all necessary tools readily available, including .Net and java development suites, Web Services are easy to design, develop, maintain, and use, compared to other integration technologies such as DCOM and CORBA.

2) Open Standards:

Web Services are based on open standards by design from W3C. This is the single most important factor that would lead to the wide adoption of web Services.

3) Flexible

Web Services based integrations are built on loose coupling between the applications that publish the services and the applications that use those services.

4) Cheap

Many EAI solutions, such as message brokers, are expensive to implement. On the other hand, web services are simple to build and would achieve the same goals in cheaper cost.

5) Scalable

Web services would allow companies to handle big applications by breaking them down into small independent logical units and build wrappers around them.

6) Efficient

As web services would allow applications to be broken down into smaller logical components, it makes web services solutions much more efficient than traditional EAI solutions.

7) Dynamic

Web Services would provide dynamic interfaces, whereas traditional EAI solutions would be static in nature.

In a paper by Gudivada, V.N. Nandigam, J. (2005), an Extensible Web Services Architecture (EWSA) was proposed. EWSA services are delivered via three channels including Web service, message queues, and .NET assembly. For each of the delivery channels, there is an adapter component residing in ServiceProvider. The ServiceProvider functions as an entry point into EWSA. It handles all service requests,

and hands them over to ProcessOrchestrator, which is in charge of processing service requests. ProcessOrchestrator is the centre of the framework; it has the requisite knowledge of all services hosted by the EWSA. ProcessOrchestrator would fulfil service requests in several steps in a specific order.

In a paper by Youcef Baghdadi (2005), it was argued web services are becoming a simple service-oriented architecture (SOA) with lower development costs. Because the underlying standards of web service allow interfacing, publishing, and binding loosely coupled services over the Internet. A Web services-oriented architecture (WSOA) was proposed. The paper also included an implementation of a specific instance of the Web services-based business interactions manager (BIM) for each category of e-commerce with regard to the business specifics.

The proposed framework consists of:

- 1) The business interactions were conceptualised by abstracting, specifying, modelling and categorizing.
- 2) The core business activities were insulated from the business interactions activities to allow the implementations of these two types of activities.
- 3) BIM functionality was specified to overcome the limitations of web services.
- 4) BIM was instantiated for each category e-commerce applications with different business specifics.

In another paper by Ricky E. Sward, Kelly J. Whitacre (2008), a multi-language service-oriented architecture using an enterprise service bus (ESB) was proposed. An ESB would provide developers a platform for the easy development of a multi-language Service-Oriented Architecture (SOA). The paper used Mule 1.4 ESB, which is an open source ESB, to communication between web services written in Java and Ada. The paper argued a system based on SOA would provide encapsulated services with well-defined interfaces for a loosely coupled architecture. The ESB would play a crucial role in SOA and connects services to applications across the network. The ESB would enable easy integration of applications of different programming languages in a SOA environment. The review has found ESB is one of the latest concept reflected in many literature as well as in the business world.

EAI platforms

Custom EAI solutions would be developed on EAI platforms. In the paper by Khubaib Ahmed Qureshi (2005), the EAI platforms identified at the time included TIBCO Active Enterprise, Mercator Enterprise Broker 5.0, Oracle Integration Server, HP Net Action IOE, IBM Websphere Business Integration Suit, Microsoft BizTalk Server 2004, Microsoft .NET, Sun Open Net Environment, Vitria Business Ware, Versata Global 2000, BEA WebLogic Integrator, Web Methods, SeeBeyond E*Gate Integrator, etc. Many of those have newer versions after the review, such as BizTalk Server and .Net platform. Most of them are easy to find on the Internet.

For example:

- 1) BEA AquaLogic was claimed to be the first service platform intended for SOA infrastructure creation and management. It allowed for development of “neutral container” in which business functions can exist independently of the infrastructure.
- 2) BEA WebLogic Server was claimed to be a recognized platform for the enterprise applications and services of Java platform.
- 3) BEA WebLogic Integration would provide EAI solutions and facilitate Service Oriented Architecture solution development.

2.3.3 EAI Quality Factors

The followings are summary of quality factors for EAI solutions from a number of papers and articles, although most of the headings are from a paper by Khubaib Ahmed Qureshi (2005).

2.3.3.1 Data Quality and Integrity

Data quality and integrity are not only the basic elements for a sound individual system; they are crucial in system integration, especially in data integration. Good data integration must protect the data quality and integrity inherited from the underlying data sources. The result data should not in any way has lower quality than those from the original source. On the other hand, only good quality data source can produce good quality data integration.

2.3.3.2 Coupling

This refers to the degree of tightness of coupling among business sub-systems. Two systems are tightly coupled if they are strongly dependent on each other. For example, one system may control the other, or they may control one another. Loosely coupled systems only interact with each other on demand. The coupling of the system must be set according to the objectives of the integration, a balanced approach have to be adopted sometimes. Tight couple tends to have quick overall responses and better performance, while loose coupling would make the integration more robust and flexible.

2.3.3.3 Flexibility and Adaptability

This refers to the ability of the integration to change and adapt itself to different sub-systems and to provide result to the ever-changing requirements. System integration is about to “add value” to the sub-systems or to achieve new goals. In the ever-changing environment like IT and the customers’ demand, good system integration must have high flexibility.

2.3.3.4 Heterogeneity

Heterogeneity refers to the degree of dissimilarity among sub-systems. The need to integrate multiple types of sub-systems has arisen due to the increased level of connectivity and increased complexity of the user requirements. Sub-systems can be different in OS platform, data type, and communication protocols. High heterogeneity would certainly increase the difficulty for system integration; on the other hand, it would also make the result of such integration more valuable once it is successfully implemented.

2.3.3.5 Autonomy

Autonomy refers to the degree of compliance of a sub system to the integration and global controlling rules. Sub systems should be autonomous in their design, communication, and execution. They only interact with each other via predefined interfaces. Sub systems very much have total local control over implementation and operation of services. They could change their processes without affecting each other. Usually, a completely autonomous collaboration may be difficult to achieve, but a

successful integration should allow as much autonomy as possible to its sub systems, by implementing adequate interfaces and translation processes.

2.3.3.6 Security

Security is always a major concern for enterprise systems, such as integration on the group of sub systems, even across organizations. Security factors must be considered before, during and after the implementation of any system integration. Tight security control on the sub systems may create more challenges for the integrated system itself. After all, adequate security measures must be applied for good system integration, no sub system's security should be jeopardized by inadequate security control from the integration, i.e. no 'back door' or a loophole should be created by the integration.

2.3.3.7 Scalability

Scalability refers to the ability of a system to grow to satisfy new requirements, which would be providing larger volume of accessible data, or higher number of transactions in a given unit of time, or handling more users concurrently. An enterprise system such as integration across systems and organization normally require high scalability, as there tends to serve larger amount of users and handle larger amount of data than other systems. Apart from hardware factors, high scalability for system integration can be achieved by using adequate interface and communication protocol. What techniques to be adopted may also include consideration of other integration factors, such as coupling and flexibility.

2.3.4 Summary of Enterprise Application Integration

The review was concentrated on Enterprise Application Integration (EAI). EAI would be regarded as a sub-set of general system integration within an enterprise network. It has become one of the hot topics since early 90s and attracted a lot of interest.

However, it was heavily influenced by ERP and its products. The review has gone through literatures to search for EAI challenges and solutions. Many frameworks had been identified including component (middleware), XML and Web Service based frameworks. Most of the frameworks were agreed to be best suited for certain purposes by their respective authors, such as workflow control. Some of the frameworks were supported by sample implementations. The review also found a number of commercial

EAI platform, including Microsoft BizTalk server. These EAI platforms are becoming commonly adopted for developing EAI solutions. The review found there would be many factors need to consider for a successful EAI solution. After all, the review found very few papers discussing the general development of EAI solution.

2.4. Development of System Integration

2.4.1 Business System Development Methods

For the purpose of this review, system integration was regarded as special kind of business system. As the definition of system integration used for the review and the rest of research, integration is for providing new features or achieving new goals by linking existing business systems together, in terms of function or data. The review had searched for papers and theories for business system development methodologies. One such paper from John Crinnion (1992) was found particular useful. The paper has summarized some trends and techniques for the development of business systems at the time, although the paper was published 17 years ago, many of its argument are still valid in the current situation.

First, the paper argued there are three major directions for development of business systems. They were described as followings:

- 1) They are towards providing greater flexibility and scalability of approach
A successful method must be able to handle systems or requirement of different kinds and different size. In the current situation, system development has been moved away from science and engineering and more towards business and management area. The approach for system development must be able to cater for issues raised from management and human behaviors, as well as from the fast changing business world.
- 2) They are towards faster development and delivery of a system
Business tend to be time and money sensitive, the development and delivery of system must be able to be fitted in for such requirements. In the current situation, more and more advanced development tools and technologies are used for speeding up system development. The Microsoft .Net development platform is one of such products commonly used for enabling fast system development.

- 3) They are towards greater involvement of the business user in all stages of the development.

In early years, systems were built to be remote from the business users. However, the climate and culture have gradually been changed; business users are becoming more and more involved in system development. In the current development world, the popular 'Agile' development method has been developed and used to allow more users involvements.

The paper also identified some new tools and techniques at the time. Some tools including CASE and 4GL are still in used but are outdated by many other newer tools now. However, some other techniques are still commonly used and very useful, they are:

- 1) **Prototyping**

It has been argued as one of most useful technique. It was originally defined as building part of the proposed system for using it to identify the real requirements. There were two types of prototyping:

Throw-away type: It is used simply to identify basic requirements and illustrate system potential. It is built in a 'quick and duty' manner and it will not be part of the final system.

Incremental (or evolutionary) type: It is built in a structured manner. It would be continually adjusted to become the full system.

The author claimed more incremental prototypes have been built than throw-away ones.

- 2) **Development Workshops**

They have been used extensively in the development world. They are gatherings of the main users, experts and decision makers for the project to derive and agree on system specification. They would speed up the communication and decision process, and would help to involve and commit the business to the project.

Proof of Concept

In Wikipedia, (retrieved October 28, 2009, from http://en.wikipedia.org/wiki/Proof_of_concept)

Proof of concept is a short realization of a certain method or idea(s) to demonstrate its

feasibility, or a demonstration in principle, whose purpose is to verify that some concept or theory is probably capable of exploitation in a useful manner. In a paper by Odysseas Pentakalos (2008), Proof-of-Concept design was described as one of the most useful development method. The paper argued the traditional waterfall approach often failed for the following reasons:

- 1) Requirements are not fully understood before the project begins.
- 2) Users only know what they want after they have seen something like an initial version of the software.
- 3) Requirements often change during development.
- 4) The effect of new tools and technologies would be unpredictable.

The paper put forward the following benefits of using Proof-of-Concept:

- 1) It would help understanding the requirements.
- 2) It would help understanding the capabilities and limitations of new technologies.
- 3) It would help assessing design decisions early in the process.
- 4) It would help the customer to visualize early on the look-and-feel of the solution.
- 5) It would reduce the overall risk of project failure.

The review found it would be difficult to separate prototype and proof-of-concept sometimes, as they would serve almost the same purposes. Proof of concept is usually considered a stage prototype development. The review found prototype would be used more often for identifying business requirements, and proof-of-concept is more often for technical aspects. Other general development methods have also been identified as followings:

1) **Incremental**

Project is broken down into small segments in order to reduce risk and to make it easy to change during the development process.

2) **Spiral**

The method combines the features of the prototyping model and the waterfall model. Each cycle involves the same sequence of steps of development including analysis, prototype, design, development and deployment.

3) **Rapid Application Development (RAD)**

It is of iterative framework type of method, same as incremental and spiral, and

consists of prototyping. The key emphasis is to fulfill business needs in the shortest time, usually use the latest development tools available. Less emphasis is put on planning in this approach. Agile development is one of such approach.

2.4.2 Integration Development Methods

Developments of system integration would be done by the same development team responsible for all the other system developments in the organization, or even on the same development platform as for other applications. However, the review has look for methodologies that particularly pertain to the development of system integration. In an early paper by David J. Schultz (1979), it was argued that there were four major approaches in developing a system integration solution.

1) **Big-Bang Integration**

The paper argued the big-bang approach was the most used approach to system integration. In such approach, individual modules are tested separately first, and then the complete system was build, tested and deployed.

Advantage:

It is good for simple and well-designed system and interface.

Disadvantage:

The testing of the complete system and interface of the modules would be difficult.

Error in the modules would be hard to identify and costly to fix.

It would create extra work for testing individual module in isolation.

2) **Bottom-Up Integration**

In this approach, low level elements of the systems were tested and integrated first, the integration then go up to higher level modules. The process repeats until the complete integration is achieved.

Advantage:

The testing of the low level interface would start early.

System errors would be easy to identify and fix.

Disadvantage:

It requires extra effort for testing.

The top level integration system is test at last. If there is design problem found at last, it will be costly to fix.

3) **Top-Down Integration**

In this approach, the systems are merged and tested from the highest level to the lowest. The method is widely used as argued by the author.

Advantage:

The testing of the high level interface would start early.

There is no need to do extra testing in isolation for the individual systems.

Disadvantage:

The integration progress would be difficult to control.

The individual system might not be ready for testing during certain stage of the development process.

4) **Mixed Integration**

This approach tries to combine the benefits of bottom-up and top-down approach. In one such approach, individual modules are tested first, and then integration will start from the top level interface. The approach would also share the disadvantages from both bottom-up and top-down approach.

Apart from the above categories, the paper also argued for the “Build” concept, which is actually “iterative development” in the current terms. Iterative development includes a series of ‘builds’ and releases; each one is developed on the base of previous one. The paper argued for a long list of advantages of the “Build” approach, some of them were extracted t here as they are still quite valid:

1) ***The top level integration would be tested early on.***

As the first iteration would be early and consists of at least some top level integration.

2) ***The testing of modules in isolation is reduced.***

As module would be integrated one by one and tested within the whole system through out the iterative process.

3) ***The critical functions would be subjected to rigorous regression testing in each build.***

Such function would be tested for each iterative release.

4) ***Users would be involved in early testing stage.***

They would be testing and using the system from the first iteration.

5) *The management would have clear picture of the development.*

Each of the iterative release would illustrate progress of the project.

The review has also found another paper by Bendik Bygstad, Peter Axel Nielsen, Bjørn Erik Munkvold (2005) on the same topic. The paper argued there were four patterns in IS integration development from the findings of the case study from a socio-technical perspective. They are:

1) **“Big Bang”**

The technology integration and stakeholder integration both happen at end of the development project.

2) **“Stakeholder Integration”**

The stakeholder integration is done step-wise, while the technology integration is done at the end of the project.

3) **“Technical Integration”**

The technology integration is done step-wise, while stakeholder integration is done at the end of the project.

4) **“Socio-Technical Integration”**

Both stakeholders and technology integration are done step-wise.

The paper has ‘Big Bang’ as the only common point as those by David J. Schultz (1979), as the papers were come from different point of view. The earlier paper was focused more system and technical issues. The later one was from a socio-technical perspective. However, the review found the ‘Big Bang’ would be one of most used term because the lack of other methodologies in the system integration area.

2.4.3 Special EAI Factors

The review has identify numerous discussion and papers on the topic of development of EAI, many of them argued that EAI is a special kind of the development with a few factors which make it different from common system development. A paper by Gian Trotta (2003) has identified the following common pitfalls of EAI:

1) **Change is constant.**

The requirements of EAI would often change, even during development, as many parts of system and business tend to be involved.

2) **EAI skills are rare.**

EAI systems would be complex and often proprietary. The combination of process, function and data issues require greater amount of skills and experience than those for a single system.

3) **Standards are never universal.**

Although much effort has been devoted on the setting up of standards by many organizations including W3C and OASIS. However, those standards are not always 100% agreed by vendors, and vendors would likely to pursue their own agenda.

4) **Thinking of EAI as a tool as opposed to a system.**

EAI would be regarded as simple tools by the management as it might not realize the complexity and involvement of the underlying systems.

5) **Treating interfaces as a science as opposed to an art.**

The EAI development would focus too much on the complex technical issues such as system interface, but not much on user interfaces. However, EAI is about link business together rather than just systems.

6) **Discarding details along the way.**

EAI system would often change. Records for system design information would become crucial for understanding the system which would be different from the original design.

7) **Unclear accountability.**

It likely to be an issue for EAI because the multi-parties and departments involvement in large EAI system.

In another paper by Nina Reiersgaard, Hilde Salvesen, Stig Nordheim, Tero Päiväranta (2005), they propose a framework aiming to provide guidance for practitioners when planning for EAI implementations. They had done the analysis for each of the three phases including chartering, project and shakedown. In the following table of the chartering phase analysis, they displayed a list of ERP and EAI factors identified from literature in column 2 and 3, the last column were finding from a case study with a Norwegian energy group. All factors were put into groups as indicated in the first column. In the finding column, “Important” indicates the factor found from several data sources. “Little attention” indicates only partial or hardly any support from the data. “New” means the factor not identified in literature survey but found important in

the data collected in this case. There were two new finding from the case study from this phase.

Group	ERP Factors	EAI Factors	EAI Case Findings
Current state analysis		<input type="checkbox"/> External pressure <input type="checkbox"/> Internal motivation <input type="checkbox"/> Existing IT infrastructure	<input type="checkbox"/> Important <input type="checkbox"/> Important <input type="checkbox"/> Important
Construct a business case	<input type="checkbox"/> Determine scope and objectives <input type="checkbox"/> Small sets of priorities <input type="checkbox"/> Set of measures <input type="checkbox"/> Identify core processes <input type="checkbox"/> Keep capability in-house	<input type="checkbox"/> Benefits, Barriers, Costs	<input type="checkbox"/> Important <input type="checkbox"/> Little attention <input type="checkbox"/> Little attention <input type="checkbox"/> Important <input type="checkbox"/> Important <input type="checkbox"/> Important
Select integration SW solution (May to some extent be performed in the project phase)		<input type="checkbox"/> IT sophistication <input type="checkbox"/> Support <input type="checkbox"/> Evaluation frameworks	<input type="checkbox"/> Important <input type="checkbox"/> Important <input type="checkbox"/> Important <input type="checkbox"/> Replacing existing systems (New) <input type="checkbox"/> Proof of concept (New)
Top management support	<input type="checkbox"/> Support and commitment from the top management		<input type="checkbox"/> Important
Commitment to change	<input type="checkbox"/> Commitment to change <input type="checkbox"/> Project champion		<input type="checkbox"/> Important <input type="checkbox"/> Important

Table 2. 2 Table 1 by Nina Reiersgaard, Hilde Salvesen, Stig Nordheim, Tero Päivärinta (2005)

The following was the analysis for the project phase.

Group	ERP Factors	EAI Case Findings
Organization		<input type="checkbox"/> Project organization (New)
Management	<input type="checkbox"/> Project management <input type="checkbox"/> Change management <input type="checkbox"/> Top management	<input type="checkbox"/> Important <input type="checkbox"/> Important <input type="checkbox"/> Important
Ownership		<input type="checkbox"/> Project ownership (New)
Skills	<input type="checkbox"/> Skills <input type="checkbox"/> Balanced team	<input type="checkbox"/> Important <input type="checkbox"/> Important
Business processes		<input type="checkbox"/> Designing new business processes (New)
Build an enterprise application integration solution	<input type="checkbox"/> Data management <input type="checkbox"/> Alignment between business and IT <input type="checkbox"/> Training <input type="checkbox"/> Testing	<input type="checkbox"/> Conceptual integration solution (New) <input type="checkbox"/> Technical integration (New) <input type="checkbox"/> Important <input type="checkbox"/> Important <input type="checkbox"/> Little attention <input type="checkbox"/> Important
Rollout	<input type="checkbox"/> Rollout	<input type="checkbox"/> Important <input type="checkbox"/> Receiving project (New)

Table 2. 3 Table 2 by Nina Reiersgaard, Hilde Salvesen, Stig Nordheim, Tero Päivärinta (2005)

The following was the analysis for the shakedown phase

Group	ERP Factors	EAI Case Findings
Address and deal with errors of previous phases	<input type="checkbox"/> Additional human, financial, and technical resources <input type="checkbox"/> Turnover <input type="checkbox"/> Top management support	<input type="checkbox"/> Important <input type="checkbox"/> Not found <input type="checkbox"/> Little attention
Make new business processes work	<input type="checkbox"/> Changes in processes and procedures <input type="checkbox"/> Retraining and additional training <input type="checkbox"/> Resistance to change	<input type="checkbox"/> Important <input type="checkbox"/> Important <input type="checkbox"/> Not found <input type="checkbox"/> GUI (New)
Handover		<input type="checkbox"/> Handover (New)

Table 2. 4 Table 3 by Nina Reiersgaard, Hilde Salvesen, Stig Nordheim, Tero Päävärinta (2005)

The result of the case study conformed to those from early literature, including the findings of the linkage of ERP and EAI. However, the case study had identified ten new factors in total from different phases, which were unique to EAI. A few of them were technical factors, such as proof of concept, technical integration and GUI.

The study use data from snapshot at a certain time during the whole project that lasted longer than a year. The factors would be found in isolation and hard to be linked to others. At last, the authors suggested more work should be done on process research to linking those factors together.

In yet another paper by Khoubati, Khalil , Themistocleous, Marinos Irani, Zahir (2006), they had identified a set of barriers of EAI adoption in a healthcare organization from a case study. They argued those barriers were also confirmed by literature review. They were:

1) **Internal Pressures**

They were from the physicians.

2) **Patients' Satisfaction**

More integrated systems would lead to higher patients' satisfaction.

3) **Evaluation Frameworks**

Selection of EAI technologies was difficult but crucial.

4) **Organizational Size**

Larger organization would need more integration.

5) **IT Sophistication**

Internal IT sophistication was an important factor for adopting EAI.

6) **Costs of EAI Adoption**

Cost was a significant factor for decision-making for the adoption of EAI.

7) **Telemedicine**

This was a particularly important sub-system in the research case.

8) **Administrators and Physicians Relationships**

Physicians should be involved during the integration process.

In summary, the review has identified a long list of special factors for EAI, most of them were either from practitioners such as Gian Trotta (2003), or from case studies research. The factors would come from many directions, including technical, management and human behavior areas. They had clearly illustrated that EAI would be very complicate and difficult.

2.4.4 Critical Success Factors

As defined by Rockart (1979, pp. 86–87), Critical Success factor (CSF) as a way of focusing CEO efforts on *‘the limited number of areas in which results, if they are satisfactory, will ensure successful competitive performance for the organization’*. The review has found the term CSF is now commonly used for IT systems and development. Numerous papers had been found and look at during the review. CSF would be very useful for identifying issues and improving results for IT projects, including EAI development.

2.4.4.1 Critical Success Factors for IT Projects

In the Standish Group (2004) CHAOS report, a list of CSFs was concluded from large-scale surveys and long time research. It was argued the data was representing over 50,000 completed IT projects (9,236 for year 2004). The report has been run for many years and was argued to be one of the most widely quoted statistics in the industry. The followings are the top ten CSFs for year 2004, although the list would change from year to year, but it was quite stable as the report said.

1) **User Involvement**

Although most developments have involved the users to various degrees, user

involvement is the number one CSF because ignoring it remains the most deadly error. It should make sure users get what they need, not just what they want.

2) ***Executive Management Support***

A project must be supported by executive management from the beginning to the end, otherwise the project will encounter great difficulties, such as lack of commitment or funding. Early deliverables, clear progress and swift finance benefits would help to ensure such support.

3) ***Clear Business Objectives***

Various stakeholders would have different goals. A commonly agreed vision should be stated clearly from the beginning of the project. It should be kept updated during the project and reviewed at each major stage.

4) ***Optimizing Scope and Requirements***

Requirements should be clearly documented and matched the business case. Scope creep would be avoided by setting clear priority and doing benefit/risk analysis for any major features.

5) ***Experienced Project Manager***

It was argued projects would be likely to succeed when there is a competent and experienced project manager.

6) ***Iterative and Agile Process***

It works like no other method, it has no formal releases but constant deliverables, updates and fixes. It was argued to be the “silver bullet” for saving project from failure.

7) ***Financial Management***

Project plans need to be associated with finance data, and they must be kept updated at all time. A system defining process and reviews at any important event would help to achieve that.

8) ***Skilled Resources***

They include competent developers, involved stakeholders and good management. Sometimes various deficiencies would be mitigated by extra training programs and staff augmentations.

9) ***Formal Methodology***

Project management must follow certain method at least track progress and

perform cost accounting. Formal project management office would be desirable.

10) ***Standard Tools and Infrastructure***

The research has found they were poorly used, but if they were used, the projects would more likely to succeed.

In a paper by Mary Sumner(1999), a case studies of seven organizations implementing enterprise-wide information management systems projects was carried out. These projects very large ones and in many cases the largest single project in the organization. For each case, the research used project data to find answers for a set of research questions, including the critical success factors for the project. In the end of the research, the following set of CSFs were summarized and presented:

1) ***Clear Project Goals***

The enterprise-wide projects must be justified with cost and benefit within the current scale of organization.

2) ***Business Integration***

The business must be prepared to change to “fit” the package, rather than trying to modify the software to “fit” current business processes.

3) ***IT Skills***

The project need to include strategies to ensure adequate skill level in IT workforce and acquire from outside if necessary.

4) ***Business Analysis***

Project analyst must have both business knowledge and technology knowledge.

5) ***Management Support***

The projects must have top management support for commitment from all involved parties and ensuring project leadership.

6) ***Training***

Projects must make commitment to train end-users for the new systems.

In summary, the paper argued large enterprise system such as ERP required centralized control, strict discipline, and extensive monitoring of project outcomes. They are different from traditional MIS projects. They focus more on standard processes and central control of the system. They are less for supporting unique business requirements at the business unit level.

Project Scope and Other Factors

In paper by Markus Biehl (2007), it was argued most of the CSFs identified in the literature were consistent with each other, but the paper had singled out project scope as a factor need to pay attention to. It is not common that project scope is included as CSF, as it is more of an attribute of the project, rather than something the development would work on or change. The paper argued CSFs would not apply equally to local and global projects. For example, top management supported would be extremely important for global projects, but not necessary as important for smaller project. The scope of project would also have direct impact on other factors including the number of people involved and complexity of communication. The paper claimed many of the cases had evidence supporting its argument. Factors like project scope and other project attributes including organization size and culture should not be excluded from consideration of CSF.

In another paper by Richard Berntsson-Svensson, Aybüke Aurum (2006), they presented a study of successful software project and products from an empirical investigation. For the study, they sent questionnaire to 15 companies (eight in Sweden and seven in Australia) and received 27 responses, as some companies provided more than one response because of variety of different projects within the company. They were all treated as independent ones for the purposes of statistical analysis. The questionnaire consisted of 33, mostly close-ended questions. Some open-ended questions were used to ascertain the relative importance of each factor. The result for software projects was summarized in the following table.

Factor	Financial services		Consulting industry		Telecommunication industry	
	Successful	Failed	Successful	Failed	Successful	Failed
Project changed PM	33%	100%	57%	50%	63%	50%
PM supported long hours	50%	50%	33%	100%	100%	100%
Staff rewarded for long hours	17%	0%	50%	100%	50%	50%
Use of specific requirements method	100%	100%	100%	100%	100%	0%
Completed and accurate requirements form the start	17%	0%	29%	0%	29%	0%
- If not, completed during project	100%	100%	100%	100%	67%	0%
Enough allocated time for requirements elicitation	67%	0%	71%	0%	50%	0%
Project's scope well defined	83%	0%	83%	50%	38%	50%
Extra personnel added to met schedule timetable	33%	100%	29%	100%	71%	100%
Commitment and support from sponsor/project champion	100%	100%	71%	100%	100%	50%
Experience project manager	90%	70%	40%	50%	80%	60%
Project manager understood the customer's problem	80%	90%	70%	60%	70%	40%
Customer involved in the project	80%	70%	70%	60%	80%	50%
Risk identification before project start	70%	40%	80%	60%	80%	50%
Good schedule estimations	60%	20%	80%	30%	60%	40%

Table 2. 5 Table 1 by Richard Berntsson-Svensson, Aybüke Aurum (2006)

The study categorized the projects into three main industries including financial service, consulting industry and telecommunication industry. The above result was then further transferred into the following table.

	Financial services	Consulting industry	Telecommunication industry
Three most important success factors	Customer involvement Committed sponsor Overall good requirements	Very good project manager Understanding customer's problem Well defined communication	Good relation between personnel Understanding customer's problem Complete and accurate requirements
Three least important success factors	Good schedule Good relation between personnel Good estimates	Good schedule Committed sponsor Overall good requirements	Good schedule Experienced project manager Committed sponsor

Table 2. 6 Table 2 by Richard Berntsson-Svensson, Aybüke Aurum (2006)

The review found the study was a mixture of qualitative and quantitative research. The questionnaire would have been sent to more companies and more investigation would have been done in selected projects. The authors argued their findings complement the results from other studies. However, they had discovered differences of those factors across different industries. They argued CSFs should be associated with project attributes such as the industry type. Similar view was identified by the review in the previous paper.

2.4.4.2 CSF for EAI & ERP Implementation

As discuss in the previous sections, EAI is different from traditional IT system and even the enterprise system like ERP, so are the CSFs for EAI. The review has identified a number of papers on the topic. In a paper by Wing Lam (2005), the author had done a case study and proposed a set of CSF especially for EAI. The paper first argued EAI projects differed from other IS projects in several ways:

- 1) They focus on integration of existing IS, not development of new one.
- 2) They require upfront strategic including what IS are to be integrated.
- 3) They impact multiple IS within an organization.
- 4) They spans across divisional boundaries, and multiple stakeholder groups would be involved.
- 5) There is not much established methodology for EAI.

Many of them are consistent with what the review had found in the previous sections.

The paper then presented a set of CSFs from literature review, which is displayed below. They are categorized in Business Organization Technology and Project (BOTP) model.

Table 1 BOTP CSF model for EAI projects

Category	Factors	Reference
Business	Strong business case for EAI (B1)	Holland & Light (1999); Sarkis & Sundarraj (2003)
	Overall integration strategy (B2)	Holland & Light (1999)
	Process interoperability with business partners (B3)	Yang & Papazoglou (2000)
Organisation	Top management support (O1)	Holland & Light (1999); Sarkis & Sundarraj (2003); Bajwa <i>et al.</i> (2004)
	Business process change and overcoming resistance to change (O2)	Holland & Light (1999); Markus (2000); Themistocleous & Irani (2001); Sarkis & Sundarraj (2003); Ross (1999); Krasner (2000)
	Good organisational and cultural fit (O3)	Sumner (2000), Krumbholz <i>et al.</i> (2000); Themistocleous & Irani (2001)
Technology	Handling legacy systems (T1)	Holland & Light (1999)
	Technology planning (T2)	Sumner (2000),
	Common data standards (T3)	Grimson <i>et al.</i> (2000);
	Use of right EAI tools (T4)	Puschmann & Alt (2004); Bajwa <i>et al.</i> (2004)
	Use of mature technology (T5)	Markus (2000); Krasner (2000)
Project	Realistic project plans and schedule (P1)	Holland & Light (1999)
	Client involvement, communication, consultation and training (P2)	Holland & Light (1999); Sumner (2000), Themistocleous & Irani (2001); Sarkis & Sundarraj (2003); Bajwa <i>et al.</i> (2004); Ross (1999)
	Required skills and expertise onboard, vendor competence (P3)	Holland & Light (1999); Sumner (2000), Themistocleous & Irani (2001); Sarkis & Sundarraj (2003); Bajwa <i>et al.</i> (2004)
	Monitoring and feedback (P4)	Holland & Light (1999); Bajwa <i>et al.</i> (2004)
	Proper migration approach (P5)	Sumner (2000)
	Adequate testing plans (P6)	Krasner (2000)

Table 2. 7 Table by Wing Lam (2005)

The case study was then carried out in a large Finance Institute (FI). In the end, the author argued the findings from the case study were largely consistent with finding from previous work. Some of the important findings from the case study were summarized as followings:

1) ***EAI is an education***

The case study revealed skills require for EAI were quite different from those for more traditional IS development projects. The organization had to hire external consultants to help complete the project.

2) ***Business integration before technology integration***

The organization had spent considerable effort to understand business integration requirements. By doing so, it ensured that technology integration was indeed for the purpose of improvements in the business process.

3) ***The enterprise data model***

If the organization had the notion of enterprise data model, the EAI project could have been done more swiftly. It was crucial for the configuration of the EAI tool and to modeling process flow.

4) ***Availability of adapters is critical in EAI tool evaluation***

Developing new adapters would be very costly, therefore the selected EAI should have all the necessary adapters or the development cost must be factored in the initial project budget.

5) ***Some custom coding may be unavoidable***

Pre-built, plug and play adapters are generally available, but as the organization had discovered some of its less-common applications which they wish to integrate had no suitable adapter available, development work was unavoidable.

6) ***Phased rollout strategy***

The organization had adopted a phased rollout instead of the simple cut-over from completely old to completely integrated systems. The rollout must taking into account of business urgency, risk management and the degree of business disruption. A phased approach is normal the most practical one.

In a paper by Zaitun A. B. and Mashkuri Yaacob (2000), they use a survey to identify CSFs using response from 40 government agencies (out of the 69 agencies requested by the survey). The survey results were processed with statistical software package SPSS version 8. Likert scale was used for collecting the data, and then the mean, standard deviation and variance were calculated. The Likert scale was from 1 (weakest) to 5 (Strongest), and factors with mean value above 3 were considered important. The following was the result from the survey:

Factors	Mean	Std. Dev.	Median
1. Support from top management	4.703	.702	5
2. Directive from top leadership	4.378	.828	5
3. Appointment of a team that can manage inter-departmental projects	4.368	.913	5
4. Establish inter-departmental coordination	4.342	.878	5
5. Complete change in work culture	3.889	1.063	4
6. Ensure sufficient measures are taken on data confidentiality and security	3.842	1.053	4
7. Outsourcing to overcome the problem of insufficient it personnel	3.816	1.036	4
8. Implementation of electronic government	3.757	.983	4
9. Setting up of an infrastructure such as the multimedia super corridor	3.750	1.052	4
10. Achieve an agreement on priorities amongst participating organization	3.684	.962	4
11. Achieve agreement on data ownership and access	3.676	.973	4
12. Open system concept	3.639	1.099	4
13. Formation of a neutral body that will act as a mediator amongst participating organizations	3.611	1.248	4
14. The availability of appropriate tools to assist in development activities	3.595	1.142	4
15. Paperless environment	3.389	1.178	4
16. Utilizing new system development methodology - such as the object oriented approach	3.351	1.086	4
17. Vendors efforts to introduce new hardware and software relevant to systems integration	3.270	1.045	3
18. EDI	3.265	.864	3
19. Agreement on project cost sharing amongst participating organizations	3.162	1.068	3
20. Complete change in leadership trend	3.139	1.073	3
21. Corporatization	2.735	1.214	3

Table 2. 8 Table by Zaitun A. B. and Mashkuri Yaacob (2000)

In the survey, the respondents were asked what were the factors that would contribute towards the success of systems integration implementation. In total, 21 factors were summarized by the author in the result table. The author then display a result from another study done by Center for Technology in Government, University of Albany in 1997, which also identified the same top CSF. The list from the previous study is as followings:

- 1) *Top management support*
- 2) *Clear purpose*
- 3) *Committed stakeholders*
- 4) *Realistic cost and benefit measures*

From the survey table, 9 technical factors were identified; the other 12 were non-technical factors. The author also calculated the difference of the two groups as following:

Pair	Total Mean	Mean of group
Technical	32.0303	3.5589
Non-technical	30.9091	2.575

As the result, the author argued non-technical factors were more important than technical ones for integration projects for government agencies, although the author had a hypothesis of a contrary view. The author also elaborated the top five factors in the table. They are summarized as followings:

1) ***Support from top management***

In the Malaysian public sector, technical innovations are implemented in a top-down manner. For any initiative from the lower level of the hierarchy, it would require a lot of explanation and convincing efforts. The author has quoted a case in one of the department.

2) ***Directive from top leadership***

It was extended from the top one. The author quoted a case of the change national identification No. and argued it would have not happen without the directive from the top leadership. Such view was reflected by many respondents.

3) ***Appointment of a team that can manage interdepartmental projects***

The author argued project implementations in the public sector tended to be very individualistic. There was not that many projects of the public sector that involve inter departmental collaborations. It would make it more difficult to succeed for integration project that would be across departments.

4) ***Establish inter-departmental coordination***

The author argued although there were many data dependency among government agencies, there has been little coordination. For any systems integration to be implemented in the public sector, it would be very necessary to first establish links between departments.

5) ***Complete change in work culture***

The author argued a complete change in work culture was quoted by many respondents. As for the time of study, government employees were orientated to focus only on accomplishing the tasks within their own department. If such tradition would not be changed, it would be very difficult to introduced new system integration or reap the full benefit of such implementation.

The review found the survey from 40 government agencies from a specific region would weaken the ability of generalization for the study result. More important, there wasn't any detail for how the respondents chosen their factors and on what base. The result of technical and non-technical grouping would be hard to use as guideline for future project or study. However, the survey results in the table certainly provide a general picture of what integration project would look like in government agencies.

In a paper by Suprateek Sarker, Allen S. Lee (2000), they presented a study using a case study to test the role of three key social enablers in ERP implementation. The three enablers were:

- 1) ERP implementation can be successful only if there is a strong and committed leadership guiding the initiative.
- 2) ERP implementation can be successful only if there is open and honest communication among the stakeholders.
- 3) ERP implementation can be successful only if the implementation team is empowered and balanced.

The above enablers were deduced from literatures by the author as the foundation for building a process theory. In order to prove or disprove these enablers, the authors used a critical embedded single-case design of the positivist case study type.

In the case study, they the bulk of data from interviews with stakeholders using an evolving protocol, which would involved more than one interview for some interviewee. Data were also gathered from direct observations, company documents, e-mails, and informal interviews to triangulate the findings.

In the case study, the ERP project was separated into three phases over an un-specified period of time. The three phases were:

- 1) Organizational structure and culture change
- 2) Implementation of core modules of the selected ERP package
- 3) Configurator implementation

The data collected for each phase including the success or failure result of the phases were then used to deductively test the enablers; the “pattern matching” method was used. The author argued the case study analysis validated the first enabler. That was strong and committed leadership at the top management level must be given significant priority for an ERP implementation project. However, the case study disproved the other two enablers, which included open and honest communication among the stakeholders; empowered and balanced implementation team. The authors also suggested satisfaction of a ‘key’ factor would compensate the lack of other factors, which they used the success of phase 1 as example, during which enabler 2 and 3 was lacking, but enabler 1 was strong.

The review found the case study was well constructed and many details of the project were discovered and presented in the paper. A positivist case study for an IT project would be difficult to execute, as there would be too many variables and the environment was usually not controlled by the researcher. Although the case proved the first enabler which was consistent with other studies, the disapproval of the other two enablers which were also deducted from literature had presented an interesting challenge for the authors.

2.4.5 Summary of System Integration Development

The review was concentrated on the development of system integration and EAI. Not many papers have been found for the topic by the review. Some of the older papers had been studied and found with still valid arguments. The review had found some general rules for software development, which were also applicable to integration development, such as iterative development and RAD. The review was then focused on Critical Success Factors (CSF) for integration projects. Many papers were found in this area and many of them adopted case study as research method. Management factors were found to be more important than technical ones by most papers. However, other factors, such as project scope, would not be ignored as suggested by some author. The review also found most research was done on the implementation of EAI products rather than solution development utilizing existing systems.

2.5. Perspectives of XML Application

2.5.1 Overview

The developments of XML applications have been gathering strength since its definition was published in 1996. In a paper by Antonia Bertolino, Jinghua Gao, Eda Marchetti, Andrea Polini (2007), it was argued that modern software systems are increasingly built according to a modular architecture. Such modularization allows the separate development of components speeds up delivery of the systems and reduces risk for the final and integrated system. Modularization also clearly requires integrations such as exchanging data in a precise and checkable format. It was argued, eXtensible Markup Language (XML) and the XML Schema had been driven by the

growth of general consensus towards adoption of standard open formats for data specifications. XML is becoming the standard format for data specification; hence make it one of the most important development areas in the computer industry. XML applications have evolved from older generation of applications such as a plain text file containing some information, which is described and tagged within the document itself, to newer generations, such as XML being used for building an entire Internet application.

These include the fact that an XML document is self described, structured data file, intended to be used by unknown group of users, such as in the Internet. Other document formats are generally intended to cater for certain customers that will typically already be known to the document creator. Creators of XML document do not need to cater for different users for the XML documents, as the documents are self described and structured in plain text. Because of this, there are virtually no limited in the ways in which XML can be used. Methods for supporting binary data in XML documents have been explored and developed.

XML is not limited for text data only. The bulk of XML applications are for Internet. With the expansion of the Internet, information expressed in XML documents can potentially be consumed by enormous customers and other business applications taking XML as input source. XML is an open source and platform independent data format; all XML applications would potentially have the entire industry as their target market. Hence, lots investment has been poured into XML application development.

XML applications are different from “traditional” applications:

- XML definition is maintained by W3C, it is open and can be extended.
- It is a format of how a document can be constructed, not a platform or programming language that an application can be developed on.
- Different users may have different visibility or usage of the information in one XML document file.

2.5.2 XML Authoring Tools & Documents

In the early day, developers could simply use XML to stored data in plain text file, as XML is an ideal format for sharing structured data among different systems. Generic

text editors and traditional methods for editing structured documents are not sufficient. New techniques must be developed or adapted to allow more users to efficiently create advanced XML documents, including visualization, analysis and transformation of data. There are many tools and product packages have been developed specially for editing XML and XSLT. The following is a limited list of products and companies producing them:

Product Name	Company Name
Authentic	Altova
Cooktop	Victor Pavlov
Emacs	GNU Emacs, Sebastian Rahtz, Christian Wittern
Epic Editor	Arbortext
Exchanger XML	Cladonia
jEdit	jEdit SourceForge project
Microsoft Office	Microsoft
Morphon	Morphon Technologies
NoteTab	Fookes Software
Open Office	OpenOffice.org, Sebastian Rahtz
Oxygen	SyncRO Soft
Stylus Studio	Progress Software Corporation
Syntax Serna	Syntax
TextPad	Helios Software Solutions
TurboXML	TIBCO Software
UltraEdit	IDM Computer Solutions
VIM	VIM SourceForge project
Xmetal	Blast Radius
XML Mind	Pixware
XML Marker	Symbol Click
XMLspy	Altova
XML Writer	Wattle Software

Most XML authoring tools rely either on schema-based templates or on the users to handle the structure of the XML manually. XML tools are developed for various purposes and usages in different fields. Many popular XML editors are text editors that

are aware of the XML syntax and follow a given DTD or XML schema. The bottom line is they ensure that well-formed documents are produced.

XML editing tools have been developed for very different types of XML documents. XML documents can be classified into two main categories: data-centric XML and document-centric XML. Data-centric XML is characterized by a regular structure. It occurs in the context of data exchange and representation of data. Document centric XML has, in general, a much more irregular structure, is often used as the means of encoding information about documents. SGML is only a very small part of XML now. XML can also be used for encoding non-textual information such as vector graphics, mathematical expressions, synchronized multimedia documents, complex forms, etc. But the main uses of XML now are for representing database, describing document standards, describing service interfaces, such as web service, and format of messages, such as for SOAP message.

The Development of XML authoring tools has produced a vast number of different products for different users in different market sectors. Some of them are heavily marketed for commercial users, such as Stylus Studio from Progress Software Corporation, others would be acquired for free or for academic usage, such as XML Marker by Symbol Click, Cooktop from Victor Pavlov. It can be foreseen that more tools and more features will be developed and put into the market, as XML is becoming more and more part of daily life for IT professional and academic.

2.5.3 XML Database

Because many enterprises have started adopting XML technology such as BizTalk from Microsoft, and ebXML from OASIS and UN/CEFACT, to process their business data, with the increasing XML usage, IT professionals are searching for solutions to effectively process and manage XML documents. Currently, there are two approaches: one is to use relational databases (RDBs) and the other is to use native XML databases (NXDs). The RDB approach seems to be used by the majority in the user community, but NXDs do exist and are being promoted and argued for both in academic and commercial world. The followings briefly cover some interesting relationship of XML data and the relational database.

The main stream of today's commercial database systems is relational database management systems (RDBMSs), such as IBM DB2, Microsoft SQL Server, and Oracle DB. Because almost all enterprises use RDBs in their information systems, it makes sense for them to use RDBs to process and manage XML data. However, there are several issues with such approach. RDBs and XML are of different structures. RDBs store data in tables and XML data are in documents with tree structure. For processing XML data, almost all database server providers offered utilities for converting data between XML and RDB formats. Examples are XML-SQL Utility developed by Oracle, Date Access Definition developed by IBM and XML ability in MS SQL server developed by Microsoft.

Many XML documents are managed in RDBMS. Storing and processing XML documents by using RDBs is mainly separated into two parts. One is to decompose and store XML documents into data tables, and the other is to retrieve data from data tables and convert them back into their original structures. This method is widely adopted in commercial databases such as Microsoft SQL and Oracle. Although the relational approach is simple and straightforward to implement, it may not provide optimal query processing performance for XML queries. But as it was argued by Eric Jui-Lin Lu , Bo-Chan Wu, Po-Yun Chuang (2003), a major the part of XML database development was XML query. They also pointed out that there are five different dimensions for XML queries including query target, query path, conditional path, operators, and functions. Most RDBMSs query engines have efficient techniques for processing equal joins for relational data, but they typically do not support those joins within XML nodes efficiently, in particular when queries involve multiple inequality-comparison predicates. Many native techniques have been developed to query XML data efficiently, they are called native approaches. It was argued by Gang Gou and Rada Chirkova (2007), the native approaches would improve XML query performance significantly because storage and query processing system were specifically tailored for XML data.

There are some practical theories and examples of combining the relational approach and the native approach. For example, XML data can be stored in the form of inverted lists by using existing relational databases, coupled with integrating efficient native

join algorithms for XML twig queries into existing relational query optimizers. In simple word, they are relational storage of XML data plus native processing of XML queries. It was argued by Gang Gou and Rada Chirkova (2007), RDBMSs would process XML twig queries more efficiently if extended relational query optimizers were adopted, other existing components of RDBMSs could also be fully reused. They argued that such integration could improve XML query processing performance and reduce system-reengineering costs. It was quoted that those current mainstream commercial RDBMSs, including IBM DB2, Microsoft SQL Server, and Oracle DB, did use such approach.

Indexes can be built in XML database. They are of the same purposes as those in relational databases. Indexes pre-built on XML data have the similar functions as those in relational databases; they are for locating goal data quickly while avoiding exhaustive scans of all the data. There are two types of XML indexes, one is of value type, which index data values in XML documents, such as classical B+-tree indexes; the other is of structural type, which index the structure of XML documents, within the structure type indexes, there are two classes of them, one is numbering schemes and the other is index graph schemes. These indexes have been used in a number of XML query processing techniques.

There are some other theories for querying XML data, such as one proposed by Hiroto Kurita, Kenji Hatano, Jun Miyazaki, and Shunsuke Uemura (2007). They argued that in order to create an efficient distributed query system for large XML data; we needed to consider three elements: data partitioning, data distribution, distributed query processing and dynamic data relocation based on both the query processing and storage costs. It was argued that data partitioning algorithm and dynamic relocation could provide more efficient query processing than the previously known alternatives.

Apart from retrieving data from target XML documents or database, XML queries must also be capable for tasks such as modifying document and data. Data modification actions are part of standard SQL language for all relational database, for XML document and database; such actions have some different meanings from relational database and have to be redefined.

Insertions: The inserted data can be simple element, complex elements, or attributes.

Deletions: The deleted data can be simple elements, complex elements, or attributes.

Modifications: - They are modifications to the values of elements or attributes.

Modifications to element and attribute names are normally not considered because they are rarely seen in data-centric applications.

According to Eric Jui-Lin Lu , Bo-Chan Wu, Po-Yun Chuang (2003), extra consideration has to be given to those actions over XML data stored in relational database:

Insertions: In data-centric applications, not only a whole document can be inserted into databases, but data can also be inserted as an element and at a specific location.

Deletions: Depending on the complexity and storage of the XML document or element to be deleted, it may involve several relational tables, many elements and files.

Modifications: Impact of modification is very similar to deletion. If modifications of element names are allowed, the structure of the documents is to be changed, and the definitions of relational database tables have to be changed then.

XML databases have some unique characteristics and challenges apart from those directly related to relational databases, such as their redundant nature of data and security models.

Redundancy

Because of their structure, XML documents have a lot of redundancy. No matter how cheap storage hardware can become, it is always desirable to have a smaller storage space, it not only make sense in hardware saving, but would also benefit in performance. In some applications (e.g., data archiving), XML documents can be compressed with a general-purpose algorithm (e.g., GZIP). There are many theories of data management for compressed XML data. One such system was proposed by Andrei Arion, Angela Bonifati, Ioana Manolescu, Andrea Pugliese (2007), which is called XQueC. XQueC's data fragmentation strategy was based on the idea of separating structure and content within an XML document, because most redundancy take place in the content part, by separating them out from the structure which normally is small in size, it allows easy identification of redundancy and application of compression on the bigger chunk of the content data. It was claimed XQueC is a full-fledged data management for compressed XML data.

Security

Security is always a predominant concern for database management. There are several discretionary access control models proposed for XML documents. Most of them allow the administrator to create security policy in a separate sheet; algorithm would be used to calculate the user's rights to the document or particular part of the document. These models mainly address the read privilege but lack of adequate address to the write privilege as most of these security models have been designed to be implemented as extensions to existing web servers. There are some other theories of XML database security model to define access control models for native XML databases, including updating on the XML documents. One of the proposals is Xupdate by Alban Gabillon (2004). Xupdate was defined by the Xupdate working group from the XML: DB initiative [XDB]. It is an XML language that can be used for updating XML documents. Xupdate Security model for NXDs has the similar functionalities of the SQL security model. It was argued that XML in its native form, not stored in a relational database, but in some variant of the Document Object Model (DOM) could provide better performance with XML documents of a very complex structure with deep nesting, and data, which are semi structured in nature.

2.5.4 XML and Data Sharing

2.5.4.1 Overview

XML has provided a lot of benefits for data sharing because of its tag based self-described construction. Contents of XML documents are intended to be understood by anyone, including computer system and human reader. Applications of XML data sharing can be on the Internet, replacing the older generation of HTML, they can also be in a Peer-to-Peer communication between two systems, or they can be for data sharing in a corporate environment on a local network. More often, providing XML data sharing over the Internet to all prospective users will have the biggest benefits of all.

In the Internet environment and transferring data to end users' browsers, XML offers many advantages over HTML. They had been nicely summarized in a paper by Len Seligman, Arnon Rosentha (2001).

- **Support for multiple views of the same content for different user groups and media.** Different user and media can easily extract content they want from the same document and present it in different ways, especially with those XML tools such as XSLT.
- **Selective (field-sensitive) queries over the Internet and intranets.** Since data content reside in individual tagged element within the document, it is easy to select targeted element when going through the entire document. This capability depends on agreements within communities on the meaning of certain widely used tags.
- **An increasingly visible semantic structure for Web information.** The structure nature of XML provides more semantic meanings for the whole document to the end users of the Internet.
- **A standard data and document interchange infrastructure.** With DTD and XML schema validation, and with those freely available parsers, common interchange infrastructure can be easily built with XML.

The uses of XML and data in such format have significantly increased in science and technology due to a desperate need for advanced technology for data. In a survey paper by Zhongyu Lu (2004), he argued XML technologies had been highly recommended to be adopted into applications in a variety of scientific domains. There were two main reasons. First, XML can be extended to develop another meta-language or tools. Typical examples are Systems Biology Markup Language SBML, MathML, CML. They all based on the built-in mechanisms of syntax and semantics like XML. Second, XML is becoming more and more popular in software development and information engineering, especially in area of integration of advanced object distributed technologies, because of open and platform independent nature of XML.

Data sharing with XML also comes with its challenges. Although XML data is in self-described format, but when it comes to integrate data from different sources, it cannot just happen naturally unless all data are in matching schemas. It was argued in a paper by Harry Halpin, Henry S. Thompson (2006), to establish compatible abstract models for all data regardless of their particular syntax would be the best way for data integration, because it would be governed by a model theory of the data using formal semantics. Regardless of whether or not such approach is feasible and desirable, it is clear that data model and schemas are the keys for XML data sharing.

The challenges of XML data integration are different from the traditional relation data. First, except for binary data, all XML data are expressed in text string, the traditional inconsistencies between string attributes, such as misspelling, will persist in the XML world and must be detected and overcome for any successful data integration; second, it is also a more important one, the structure of the XML documents have created additional complexities. As it was argued by Sudipto Guha, H. V. Jagadish, Nick Koudas, Divesh Srivastava, Ting Yu (2006), autonomous data sources might contain the same data but they would have different structure. It is important to be able to correlate such data, as they are the same, but because they may or may not have optional elements and attributes, they can not be matched exactly. Even when data sources have the same DTD and XML Schema, such situations can still happen.

2.5.4.2 Data Sharing Frameworks

A lot of papers and theories have been devoted to the challenges of XML data integration and data sharing. Some of their solutions are summarized as followings:

Metrics and Algorithms

It was proposed by Sudipto Guha, H. V. Jagadish, Nick Koudas, Divesh Srivastava, Ting Yu (2006). It argued that joins are to be created between XML documents for integration. The match in structure and content of XML documents can be quantified and defined with notions of distance that have metric properties, which in term can be used in the join operation or incorporated in a framework to facilitate the join operation. They have also gone further to propose algorithms to fine-tune the performance of such joining operations. To facilitate this operation, the new idea of reference sets was introduced, which consists of data elements used to project the data space. Reference sets are used by the algorithms to calculate joins and have big impact on the performance. It was also argued their framework was vastly different from structural joins. Structure joins are used to match XML documents to XPath queries based on structural equality, as opposed to approximately matching XML document fragments based on structural difference. Structure joins are commonly applied during XML query answering.

Integrated Information Retrieval (IIR)

Bremer and Gertz (2006) introduced integrated information retrieval (IIR), a conceptually new approach to integrated data and document retrieval based on XML. They introduced the syntax and semantics of an extension of the XQuery language called XQuery/IR. In IIR, it goes through arbitrary, intermediate sequences of document fragments (DFs) to rank and retrieve document. All instructions are embedded into an XML query language. In particular, IIR provides for a meaningful nesting of data and document retrieval sub queries, which allows for answering new kinds of queries. In summary, it was claimed that XML data retrieval and integration can be achieved in the same way as for traditional relational data by using IIR, and XQuery/IR.

Mediator-based Data Integration

There are many theories falling in this category. One of such theories was proposed by Bernadette Farias Lóscio, Thiago Costa, Ana Carolina Salgado (2006). It was argued that at the centre of the integrating data from different sources, there is a need to provide the user with a virtual view called global schema independent from the sources. The main objective of all mediator-based data integration systems is to provide a unified view of several distributed and heterogeneous data sources. Such mediation schema will correspond to a set of elements on the local data sources. The main challenge of such approach is to re-formulate the queries on the global view into the source schemas. They proposed X-Entity, an ER-based conceptual data model, and XEQ, an XML-based language, whose specification facilitates both query decomposition and translation.

XClust

The theory was proposed by Mong Li Lee, Liang Huai Yang, Wynne Hsu, Xia Yang (2002). XClust is about clustering the schemas (DTDs) of XML sources to make integration more effective and scalable. It claimed that because reconciling similar DTDs within a cluster is a much easier task than reconciling DTDs that are different in structure and semantics. XClust assesses the similarity between DTDs according to their semantics, immediate descendents and leaf-context similarity of DTD elements, and then store those sources into clusters according to their similarity. In summary, XClust works by simplifying the source schemas and matching element by name and

its position in the schema. It was argued that XClust could offer significant advantage for large-scale integration of XML sources, because XClust could produce the best DTD element mappings. XClust could also guide the integration process by telling how similar the documents are for integration, if they are of little similarity, the integration should be simply stopped.

Intentional XML Data

The theory was proposed by Tova Milo, Serge Abiteboul, Bernd Amann, Omar Benjelloun, Fred Dang Ngoc (2005). It was established on the fact that in the current Internet-connected world, almost all up-to-date data and document can be retrieved instantly from the web. It is a matter of fact that more and more web pages consist only links to the sources of data they are displaying, the data itself are retrieved only when the pages are displayed. They introduced a new type of XML documents that they called intentional documents. In these are XML documents, some of the data is given explicitly but others are defined only intentionally by embedded calls to outside web services. Integration and exchange of these documents bring in special challenges, such as whether or not to materialize the intentional data (i.e., to invoke the embedded calls) before the document is dealt with. It was argued that schemas (DTD and XML Schema) could be used to control the exchange of intentional data and, in particular, to determine which data should be materialized before further action, and which should not.

2.5.4.3 XML Data Sharing for E-government

Data sharing is important for most E-Government initiatives. It can form the core part of any cooperation and integration within a government agency or among them. There are many theories and framework for E-government based on data sharing and XML.

In a paper by Zellou, A.; Chiadmi, D.; Front, A.; Giraudin, J. P (2004), they proposed an E-government architecture based on three major components, first is the portal for single access point, second is the mediation for integrations and the last is XML for data sharing and exchange. They argued that a common language like XML is an essential part of any data sharing and E-government architecture.

In a paper by G.M. Bryan, J.M. Curry, C. McGregor, D. Holdsworth and R. Sharply (2002), they proposed to use XML to facilitate information management across multiple local government agencies. They argued the main issue of data sharing is lack of common data definition and common database platform. Such issue must be resolved as there are more and more requirements for data interchange in the modern IT industry, especially among government agencies. With such a large variety of database systems in use and the way these systems are implemented and used, they argued that the solution would only be an XML based system, which would allow collaborations among those existing systems. They implemented a research project of a fully functioning XML-based prototype system that provides integrated services from a group of legacy systems. They claimed their system was of open standard and based on heterogeneous database and server platforms; it could also provide service over the Internet. Their system was basically a data sharing broker built with XML, XSLT and middleware technologies.

In a paper by Ivor Perry (2003), he argued the reasons why XML is the primary choice for data sharing are for its ease of deployment and its cost effectiveness. He proposed to use XML to build a workflow management system in the health care section for the UK government. He argued that the reasons for most E-government projects to choose XML are not because of any 'political correctness', there are indeed very good reasons. In term of technologies, XML is in plain text and seems to be easy to learn. It will allow government agencies to do in-house development and maintenance. There are a plenty of XML tools available so it makes it cost effective to use XML. He argued that his theory had been proven by that there were so many workflow management systems out there developed in XML.

In a paper by Wen He, Peng Lu, Chao Zhou, Xuelian Lin (2006), they proposed a web based E-government information sharing system. Their project was based on a web service, P2P and XML architecture. They argued data sharing could enhance the collaboration of internal departments and interactions between government agencies; it could also reduce or eliminate unnecessary data duplications and associated cost. They chose the technologies because they had proven benefits of scalability, efficiency and accuracy.

In a paper by Airi Salminen (2005), she had summarized system integration and data sharing in the following classes:

- A common user interface to access all underlying systems
- A portal to provide directions and linkages to various systems
- Internal data exchange between software systems
- External data exchange between software systems

She argued that in all these cases XML could offer syntax and meta-language for developing a common format for data exchange. Her paper was focused on building digital government, especially for European countries. Europe has large number of nations and different languages, but with the expansion of the European Union, it becomes more and more necessary and important to improve data exchange and system collaborations. Some areas such as custom and police departments are in urgent needs for these data sharing and system integrations. They are also needed for making use of those legacy systems from those new countries joining into the union.

One of the early forms of data sharing among different systems is by electronic data interchange (EDI) standards. One of the early one is the EDI connection between the Finnish Parliament and a publishing house, it was created in the beginning of 1990's. However, these traditional methods are gradually replaced by XML-based formats both in commerce and in public administration. In Finland, the government has a portal at Suomi.fi in the Internet, it classifies and describes various public services and provides links to them. According to the paper, the portal data is updated daily automatically by connected systems of the county union and various government departments, all data exchange is in XML format. In UK, the e-Government Schema Guidelines for XML have proposed the e-Government Interoperability Framework (e-GIF) for the UK government. It also specifies XML as the primary means for data integration. This is driving the ongoing development of XML schemas for government systems. These XML schemas will become the base for future data exchange and system integrations.

2.5.5 XML and System Integration

Most modern day organisations have multiple information systems running on their networks and they are would be all essential to their missions. There is a trend toward

combining these computing services and creating interoperation among them for the benefits of scalability, flexibility, and efficient resource utilization. To exchange data elements based on a common data interpretation among those systems can be one of the biggest challenges for any system interoperation.

2.5.5.1 XML Frameworks

The review had identified many theories for using XML in system interoperations. Some of them are summarized as followings.

XML-enabled Workflow Management

It was proposed by German Shegalov, Michael Gillmann, Gerhard Weikum (2001). It was a system framework for leveraging XML technology for Internet-wide workflow management. The concept is to transfer all activity calls into special kinds of XML documents, an XML mediator system is to be used to manage the flow between the workflow engine and the activities, all communications are in HTTP protocol allowing the whole integration to be run over the Internet. The XML mediator carries out all the cross talk among components with all data exchanges in XML documents.

Components, business-object servers, or workflow servers, do not need to make any changes or to know any details about the other parties. In summary, the proposed XML mediator application encapsulates all server applications' interfaces through e-service clients and into XML format. Only the XML mediator itself needs to cope with the full heterogeneity of different workflow engines, it provides unified interfaces to all types of activities from those engines. New workflows and activities can then created and maintained through the client interfaces of the XML mediator.

XML Schema Extension

In XML documents, the formats of messages and the names of the tags are specified in a *schema*, which itself is expressed in XML. The extension theory was proposed by Johannes Helander (2004). The key concept of the schema extension is to include special elements in the XML documents for accommodating future changes and new elements for the XML document. For example, it was proposed in SOAP messages, an explicit 'mustUnderstand' attribute is to be included and used for controlling what extensions can or cannot be ignored. It was argued that such properties made XML more useful and robust in heterogeneous environments where different computers use different software and different versions. With the schema extension theory, the

schema describing the XML data is flexible and no necessary the same for receiving applications to be able to understand those only necessary data and react in the ways they are supposed to. With only the minimum and 'mustUnderstand' part of the schema and data being the basic requirement for the interoperation, it has made the process much easier and more seamless. The theory was mainly targeted for devices with embedded applications.

X-GTRBAC

It is an XML-based policy specification language proposed by Rafae Bhatti, Arif Ghafoor, Elisa Bertino, James B. D. Joshi (2005). They claimed that X-GTRBAC was a specification language based on the GTRBAC model that incorporates the content- and context-aware dynamic access control requirements of an enterprise. It was targeted for building access control for enterprise-wide systems. GTRBAC model is a generalized extension of the widely accepted role based access control (RBAC) model, which can be used for defining a diverse set of access control policies. RBAC model can simplify authorization administration in large enterprises. A Java implementation has also been presented by the authors. One of the reasons for adding XML to those existing model is because XML provides a uniform, vendor-neutral representation of enterprise data. The new model would allow interchange, sharing and dissemination of information content across heterogeneous systems. It was argued that XML was a natural choice as the basis for the enterprise policy specification language, because its syntax and semantics can easily accommodate the unique requirements of an enterprise. The widespread support of XML from all the main platform and tool vendors also gives XML extra advantages for promoting interoperability. The X-GTRBAC can be deployed as both stand-alone and web based application, so it may be invoked from either the local system, or remotely through XML aware browsers.

Binary Format XML

Jaakko Kangasharju, Sasu Tarkoma (2007) argued for using binary format for serializing (transmitting) XML data between applications. Because XML as a format is very verbose, it was very difficult to adopt XML in areas where compact data representation is important. In another case, there are concerns of efficiency when high-performance must be achieved and a large amount of binary data is processed. The binary format standard for XML was actually drafted by XML Binary

Characterization (XBC) Working Group chartered by the W3C. It was argued a widely accepted binary format for XML would be useful in bringing such new areas to the XML world from the start. In the authors' paper, they presented two experiments to demonstrate the benefits for adopting binary format XML. One is for comparing parsing speed of several XML parsers against a fast binary format parser using several different data sources. The other is for comparing decoding with binary format data and text format data. It was argued that binary format is especially beneficial for science computing.

2.5.5.2 System Integration in E-government

For New Zealand government, there is an E-government Interoperability Framework (e-GIF) established. It is a set of policies, technical standards, and guidelines. It covers ways to achieve interoperability of public sector data and information resources, information and communications technology (ICT), and electronic business processes. It enables any agency to join its information, ICT or processes with those of any other agency using a predetermined framework based on "open" (i.e. non-proprietary) international standards. It was argued that the e-GIF performs the same function in e-government as the Road Code does on the highways. Most of the standards in the New Zealand e-GIF are based on the interchange format, XML.

One major objective of E-government is interoperation within and among government agencies and their IT systems. Government agencies are always among the biggest users of IT systems anywhere around the world, so they have and accumulate a big number of existing systems. How to make the most of these existing systems and adapt them to fulfill new requirements are always a priority and a challenge. System integration is about bringing together those existing systems and providing new functionalities in the most cost effective way. It is also the only way for some government agencies that have complex problem to deal with in the modern world. In a paper by Seema Degwekar, Jeff DePree, Howard Beck, Carla S. Thomas, Stanley Y. W. Su (2007), they proposed an event-triggered data and knowledge sharing among collaborating government organizations. They argued that government agencies of today are facing complex global problems such as border control, illegal immigration, terrorism, bio-security threats, among others. Effective collaboration amongst these agencies holds the key for solving these complex problems.

Effective data sharing is the foundation for system integrations for government agencies. It is not enough that the sharing only include some basic and small amount of data in simple forms, more sophisticated and large amount of data sharing are essential for many new requirements. These sharing may include human and organizational knowledge useful for decision support, problem solving and activity coordination. They also require new and effective interfaces for communication and transformation in many cases, as data are from distributed and heterogeneous sources. It was argued that schema mapping are important for achieving meaningful sharing of data.

2.5.6 XML and Web Service

2.5.6.1 Overview

A Web Service is defined by the W3C as "a software system designed to support interoperable Machine to Machine interaction over a network." Web services are designed as web based programming interfaces that can be accessed over a network, especially the Internet. They are executed on the hosting servers. Client applications submit requests and receive result of the execution. Communication between the web service clients and servers commonly use XML messages that follow the SOAP standard. SOAP stands for Simple Object Access Protocol, is a lightweight XML-based messaging protocol used to encode the information in Web service request and response messages before sending them over a network. There are also machine-readable descriptions in XML-formatted language of the operations offered by the service written in the Web Services Description Language (WSDL) provided by web service servers. They are prerequisites for automated client-side code generation in many Java and .NET SOAP frameworks (frameworks such as Spring, Apache Axis2 and Apache CXF being notable exceptions). In summary, as argued by Marlon Pierce, Geoffrey Fox, Choonhan Youn, Steve Mock, Kurt Mueller, Ozgur Balsoy (2002), *"Essentially, Web Services are an XML-based distributed service system."*

2.5.6.2 Advantages

Web services allow different systems to communicate with each other easily, because all communication is in XML, Web services are not tied to any one platform. For instance, Windows applications can talk with UNIX applications, Java can talk with

PERL. Web service has pushed the information system development a lot further after it has been evaluated from traditional client-server and then web based thin client phases. More and more new system developments use the web service architecture as long as there are requirements for any client server communication and distributed process. The current momentum is also owed to large software suppliers such as Microsoft, IBM and Sun. An ample of new software development tools in .Net and Java from those suppliers have made the use and development web service based application so much easier than any other type of development before. In fact, XML web service has dominated the distributed application development since so many powerful .Net and Java tools were marketed and promoted by those big software suppliers.

2.5.6.3 Disadvantages

Although advantages and benefits of XML web service are so obvious and profound, there are some disadvantages, which we may also have to be aware of:

Performance

Because of the verbose XML format, Using XML web service can be considerably slower than traditional client-server protocol. The issue can be worse when large messages have to be passed.

Lack of client notification

When XML web service only use HTTP protocol, client cannot be notified by server, while notification is standard feature in traditional client server situation.

Lack of versatility

Original XML web services are designed for providing simple form of service invocation. They are not supposed for a lot of supporting services (such as persistency, lifecycle management, transactions, etc.) like the traditional client server relationship. Although new service specifications and development tools were added for improve its ability in such areas, they require extra work and overhead.

2.5.6.4 Special Aspects

Apart from the general system development for supporting the day-to-day business and organizations, there are many theories on how XML web service can be used in special environments or for achieving special goals. Some of them are summarized as followings:

Benchmarks for Grid Web Services

The benchmark framework was proposed by Michael R. Head, Robert van Engelen, Wei Zhang (2006). It was argued in the Grid Web services landscape, there was a lack of fundamental metrics and micro-benchmarks for Web services based grid middleware, such benchmarks are crucial for providing insights on performance limitations, bottlenecks, and opportunities for optimisations. It was claimed the proposed framework would help evaluate and provide a road-map for the evolution of the architecture and design of grid middleware. For Web services based grid middleware, it would also provide insights to various performance aspects.

Semantic fXML

Semantic fXML was proposed by Harry Halpin, Henry S. Thompson (2006). They claimed that *“It is a paradigm for uniting the diverse strands of XML-based Web technologies by allowing them to be incorporated within a single document. This overcomes the distinction between programs and data to make XML truly self-describing.”* Semantic fXML is a XML vocabulary that can be used to describe web service as embedded functions within XML documents. It was argued that the benefits for such XML documents are: first, it could allow the XML document to be parsed and break down into functions and data, and then allow them to be processed and computed separately; second, with such XML document making up of the Internet, the entire Internet could then be computable and the data integration could be seamless.

Security

A paper authored by Ernesto Damiani, Sabrina De Capitani di Vimercati, Pierangela Samarati (2002) summarized some of the issues and solutions surrounding security of XML web service. It was argued, the original specifications of their underlying technologies for XML did not even mention security. It was therefore why security is one of the biggest concerns for future development of XML web services. They pointed out that there were two major issues regarding security of XML web service. They were restricting access to a XML Web service to authorized use and protecting the integrity and credential of XML messages exchanged in a Web service environment. It was argued that, apart from the protocol-level such as HTTPS, which would add overhead to the base line communication, it was preferred that message level signature and encryption should be used. The authors claimed security for SOAP messaging could be successfully addressed by applying signature and encryption

standards. In summary, security has been addressed from a number of directions. The main one is the SOAP message signature specification. Security control the message body can be specified in the message header. The header can specify credential transfer and access control for the message body. Apart from the SOAP signature, there are also XML security specification language, such as security assertion markup language (SAML) and eXtensible access control markup language (XACML).

2.5.7 XML Summary

2.5.7.1 Advantages

In a paper by Brahim Medjahed, Boualem Benatallah, Athman Bouguettaya, Anne H. H. Ngu, Ahmed K. Elmagarmid (2003), They argued that a large number of contemporary B2B interaction standards are based on XML, XML was becoming an emerging standard for data representation and exchange on the Internet. They argued one of biggest advantages offered by using XML was that it is much cost effective than other means such as traditional EDI and cost associated with them. There are many commercial and public XML processing and integration tools. The cost in an XML based approach is to be less significant than other approaches because of widely available XML processing tools.

In a paper by Wolfgang Emmerich, Ernst Ellmer, Henry Fieglein (2001), they argued some important advantages of using XML are the availability of standards of XML data in various industry sectors, the large number of evolving supporting tools and more and more XML based interface support from product vendors. They proposed an XML based middleware architecture called TIGRA. They argued that their middleware component and XML are complementary to each other. By using XML, their architecture will enable reliable transport of data between multiple distributed systems. As XML can also express the structure of data, so semantic data can be expressed and transferred appropriately by using available tools. The XML interfaces of their framework and systems provided by other suppliers also make it simple for future integrations.

In a paper by Zachary G. Ives, A. Y. Halevy, D. S. Weld (2002), they argued that XML had become the 'lingua franca' for data exchange and integration across

administrative and enterprise boundaries. Almost all database engines and data providers have offered XML features. There were wide ranges of standard XML Schemas and DTDs for different sectors in the IT industry. They argued that the biggest advantage of XML was that it has removed one of the major obstacles to integrating data from widely disparate sources - namely, the heterogeneity of data formats. They proposed an XML data integration system called Tukwila, which is an XML query processor in core. They argued, with their XML data integration system, the advantages of using XML could be further extended in the following areas:

- *The ability to query, combine, and restructure the content of XML documents of arbitrary size.*
- *The ability to combine data from multiple sources, including data that is the result of dynamically computed queries.*
- *Support for a “streaming” or pipelined query processing model that produces results as soon as possible. (Zachary G. Ives, A. Y. Halevy, D. S. Weld (2002))*

In summary, XML has opened up a virtually unlimited data source, from different platform, format and source. In the age of Internet and connected computing, the biggest advantage of XML would be that it enables us to access data in a lot more effective and efficient ways.

2.5.7.2 Disadvantages

The main disadvantages of XML are inherent in core features of XML. As it is text based, it uses more space than binary format for storing the same amount of data, especially in data type like numeric and date time. It is also flexible in format although XML schema and DTD can be applied; it is much less compact and rigid than relational data. The last and also the most important feature of XML is that it is self-describing with embedded tag for every data element. For any large piece of data or record set, those repeated tagging carry a huge amount of redundant information. All these characteristics of XML would be human friendly and built-in for what it was invented for, data publishing, but they are not for computers and efficiency. One of the biggest disadvantages of XML is its low efficiency. To improve performance for XML parsing is one of the biggest challenges.

In a paper by Margaret G. Kostoulas, Morris Matsa, Noah Mendelsohn, Eric Perkins, Abraham Heifets, Martha Mercaldi (2006), they proposed a XML processor called

'XML Screamer' aiming to provide high performance in XML parsing with an integrated approach. They argued that with careful attention to processor implementation, API design, and application integration, XML could in fact be processed much more rapidly than common practice would suggest. They claimed their XML Screamer was faster than most available processors, and it demonstrated that XML Schema validation could be done at similarly high speeds. After all, performance issue of its processing still remains to be a disadvantage for XML.

Querying XML data may also present some challenges, especially when they are compared with the relational database system. However, since most commercial relational database systems like Oracle and Microsoft SQL office these XML functions, these issues are becoming less important and significant.

Storing data in XML still only happens in relatively a very small area in the current IT world. No significant break through technique has been identified so far and it would remain so for the near future. The majority of XML applications, especially for E-government are focused in data sharing and system integrations, so any issue regarding to XML storage is not relevant or very minor.

2.5.8 Summary of XML Applications

Although XML had been mentioned in the previous chapters, the review had further explored XML and attempted to cover a wider area of XML application. It could be concluded that XML was not just accidentally involved in system integrations; XML would be the most important elements in many system development especially in integration with very good reasons. Since the first definition of XML, research and development effort, achievements and challenges have gathered more and more momentum. In fact, it have been developed into a family of technologies, such as XML Information Set, XML Schema, XML Query, The Extensible Style-sheet Transformation Language (XSLT), XLink, Xpointer, XML Forms, XML Protocol, XML Encryption, XML Signature. XML applications would be ranged from XML editing, XML document, XML database, data sharing, system integration and web service. Because of the self-described text nature of XML and all the benefits coming with it, XML has huge advantages over other data format, and XML has been adopted

in many system development areas including system integration. The review had found numerous XML applications in this area, most of them were for data sharing and web service based integration frameworks. Many of the theories were based on practical application for a particular purpose such as data sharing and workflow control, and then a framework or a solution was proposed for general purpose. The review found these theories and frameworks very difficult to generalize. Further search in this area also revealed very few theory or framework is widely accepted. However, the review found numerous papers argued for the benefits of XML, it was generally agreed that XML would be very useful for any system integration development, regardless of any drawback it may has or any particular approach is adopted.

2.7. Review Conclusions

2.7.1 Review Findings

The following findings have been deduced from the literature review:

- 1) System integration has been an important topic and has attached much research and development effort. There are numerous products in the commercial world and the similar amount of theories and frameworks proposed in the academic world.
- 2) Enterprise application integration (EAI) has become one of the hot topics since early 90s. There are large amount of product and theory found by the review.
- 3) The review had found only a small number of research methodologies for system integration and EAI. Many paper found in the review were based on case study.
- 4) The review had found a large amount of literature on the evaluation of IS implementation, system integration and EAI, many of them focused on critical success factors for the projects.
- 5) The review had found many top critical success factors were management related, such as top management support, project management and user involvement.
- 6) The review had found most of the theories regarding to integration were about implementing package product such as ERP and EAI software, only small number of paper was found researching on the development of integration solution.
- 7) The review had found extensive use of XML in the ICT industry for various purposes, including XML documents and databases, integration frameworks and other e-government initiatives. However, apart from those frameworks or theories,

the review found there is a lack of useful discussion or guidelines for XML application in system integration development.

- 8) Case study method had been found widely adopted for IS research in many paper covering many IS topics, including system integration and EAI. The method had been found particularly useful in studying integration development method and identifying critical success factors.

2.7.2 Research Questions

The findings have led to the following questions; the answers for them are to be found in this research project.

- 1) *What are the core issues for system integration from system development perspective within large organisations such as government agencies?*

- 2) *What are the system integration development techniques, particularly XML applications, for resolving these issues?*

Chapter 3 Research Proposal and Method

3.1 Research Questions

3.1.1 Question List

The following research questions were derived from the questions from the literature review:

- ❖ *What are the core issues for system integration from system development perspective within large organisations such as government agencies?*
 - 1) What would be the system integrations within a large government agency?
 - 2) What were the integration solutions?
 - 3) What would be the critical success factors of these integrations?
- ❖ *What are the system integration development techniques, particularly XML applications, for resolving these issues?*
 - 4) What would be the methods for resolving these issues?
 - 5) What would be the XML applications for these integrations?

3.1.2 Question Explanation

1) What would be the system integrations within a large government agency?

There would be many varieties of IS integrations within organizations who may own and operate numerous systems. Systems integration is different thing to different people. For example, in an early paper by Nilsson, E.G.; Nordhagen, E.K.; Oftedal, G. (1990), the authors argued that there were four main areas of system integrations, each focuses one aspect of integration and could be addressed independently off the others. They are integration technology, integration architecture, semantic integration, and user integration. After reviewing a large amount of literature, the review concluded that there were a few areas including data integration, functional integration and workflow integration had attracted the most attention and effort in the recent years.

Integrations are becoming more and more important as organizations have added more and more systems into their portfolios. In the past ten years, as Internet and other computer technologies have evolved, it has brought even more challenges and excitements to system integration researchers and practitioners. A lot of system

integrations have been achieved using the latest technologies and brought great benefit to users, especially in B2B commercial applications. However, system integrations within a large organization would have many unique characteristics different from those B2B frameworks. Currently there are many theories including EAI, ERP and workflow integration trying to cover topics in this area. All those theories would only be proven in the research aiming to answer the above question.

2) What were the integration solutions?

IT system development has evolved from basic software engineering projects to management systems of complex business process. As it was argued by John Crinnion (1992), there were two paradigms for IT system development, one viewed it as a branch of science and engineering, the other is primarily concerned with the management of change in human-activity systems. It was argued the later approach is being more and more adopted. Work is still to be done to identify the difference of management of integration projects from common system development, as very few theories have been found in the literature review that were focused on the development of integration project. However, in a paper by Lam (2005), the author argued integration project would be similar to ERP implementation. The investigation for answering the research question will explore in this area.

3) What would be the critical success factors of these integrations?

Integration development is highly complex that can be addressed from many directions. One of commonly used perspective is critical success factors, especially from project management point of view. According to a paper by Lam (2005), critical success factors for system integration mainly include management elements such as top management support, overall integration strategy, realistic project plans and schedule, client involvement and etc. Not only critical success factors would affect each other, they would also be affected by other non critical success factors, such as project scope.

4) What would be the methods for resolving these issues?

A lot of technical integration theories and frameworks in commercial and academic world were identified in the literature review. Most of papers and theories for system integration are for general systems across multiple organizations or with specific

industry focus, such as in E-commerce and cross government agency areas. There is a lack of attention and focus on integrations development within organization boundary. The reasons for such phenomenon may include that each organization has its own unique characteristics and research on a single organization would not generate sufficient interest or guarantee generic solutions. For this research, the researcher would argue if we can have an in-depth research on a single agency, we can draw out some conclusions that can be useful in general situations. A research for identifying methods to resolve issues for system integration within large organizations, such as government agencies would be very beneficial.

5) What would be the XML applications for these integrations?

For many years, XML has become a hot topic for research and development in both academic and commercial world. In the literature review, numerous XML applications were found in the many applications, such as XML documents, XML authoring tools, XML database management, data sharing, web service and content management. XML applications in E-government and system integration were among those most challenging areas. It has been found in many theories and frameworks that there are many ways in which XML can bring benefits to integration solutions, such as being used as data format for sharing data among systems, or being embedded in web service SOAP procedures calls for system interactions. As it was argued by Margaret G. Kostoulas, Morris Matsa, Noah Mendelsohn, Eric Perkins, Abraham Heifets, Martha Mercaldi (2006), XML is widely accepted as a means of exchanging structured information because of its ability of structure and self-describing. This research was to identify exactly how these would happen and what methods could be used in the most appropriate way. It was also found system integration had have a much longer history than XML, as argued in an early paper by Nilsson, E.G.; Nordhagen, E.K.; Oftedal, G. (1990). There are many integration solutions implemented without XML at all. Since XML was formally defined in 1996, it has been attracting more and more interest and showing strong development growth. More and more integrations have made use of XML and take advantages of its benefits. In the review, a large number of theories were found proposing to use XML for system integration; it was argued that XML could provide a lot of advantages for implementing data sharing and system interaction. In a paper by Brahim Medjahed, Boualem Benatallah, Athman

Bouguettaya, Anne H. H. Ngu, Ahmed K. Elmagarmid (2003), the authors argued using XML is the most cost effective approach for system integration.

Although many advantages and disadvantages for using XML for integration were identified in the literature review, the review has found a lack of research or practical applications of the XML-based integrations in organizations such as government agencies. This research would help exploring and providing clearer pictures in this area with evidences identified from the study. As stated previously for the research objectives, the research was aimed to come up with arguments for any theory or framework on how XML can be adopted in system integration, or provide directions for future research and development.

3.2 Research Methodology

Methodology selection was important and had been a challenge for this research. A literature review on IS research method has also been conducted and attached in appendix E. The research questions were the primary factors for choosing a suitable research method. All research questions are of exploration nature starting with “What”. The questions have particular interest in system development issues with technical focus, which would require special consideration.

3.2.1 Preliminary Evaluation

Epistemology Classification

Positivist Research

As from the literature review, a positivistic research is to gather evidences for formal propositions. It would not be appropriate for this research as the research was not to prove any propositions but to investigate and identify issues and solutions of system integrations in a government agency.

Interpretive Research

As from the review, an interpretive research is to try to give explanations or to understand phenomena through accessing the meanings participants assign to them. As all the research questions listed above are started with “What”, they are designed to explore or understand phenomena for certain situations, in this search, they were

system integrations in a government agency. This research should be an interpretive one.

Critical Research

As from the review, critical research is to aim for making changes and helping to eliminate the causes of unwarranted alienation. From the research questions, there is no intension of making any change on any things, such as system models or development methods. This research was mainly aimed for understanding system integrations and hence would not be a critical research.

Quantitative & Qualitative Research

Quantitative Research

As from the review, quantitative research is a process of collecting of numerical data in order to explain, predict and/or control phenomena of interest. The research target would be issues related to system integration within the selected government agency, there would not be any large quantity of numerical data involved, which had ruled out the possibility of using quantitative method for this research.

Qualitative Research

As from the review, in qualitative research, data is collected in various ways including observation, interviews and questionnaires, documents and text. Researchers would then explain, predict and/or control phenomena of interest. The research questions were clearly for understanding and explaining issues around system integration, which had led to the selection of qualitative method for this research.

3.2.1 Prospective Methods

From the preliminary evaluation above, it was concluded the methodology for this research should be an interpretive research of a qualitative type. According to a paper by M. R. de Villiers (2005), research methods fell into this category include observation, surveys, questionnaires, interviews, case study, focus group, ethnography, documents and artefacts studies. The author also proposed another three main type of interpretive research methods for IS, they were development research, action research, and grounded theory. In another paper by Robert L. Glass, V. Ramesh, Iris Vessey (2004), the authors had done a survey of research method in Computer Science (CS), Software Engineer (SE) and Information System (IS), those methods used by majority

of IS research were: field study, laboratory experiment, conceptual analysis, and case study. All these would be methods for this research, but they are of a large number and unrealistic to be evaluated one by one. In the following section, a selection of those methods that had been considered for the research are described.

Interview

An interview is a conversation during which one person (interviewer) asks questions and another person (interviewee) answers them, it would also happen between multiple people. It is a direct and effective way to understand the interviewees and issues related to them. It has been widely used in IS research.

The advantages of interviews related to this research would be:

- 1) It would enable the researcher to engage with interviewees directly and have direct feelings of the issues.
- 2) It would allow interviewees to explain issues in details and elaborate any opinion they would have.
- 3) It would allow the researcher to explore further on the issues based on previous answers.
- 4) It is a quick and direct way to understand complex issues, especially those related to human behaviours, organization and management.

However the disadvantages are:

- 1) Interviewees must be carefully selected and be available.
- 2) A reasonable number of interviewees are needed to ensure the validity of the findings and the research.
- 3) The interviewees have to be able to recall details of projects that would have happened some time ago.

Interview was regarded to be ineffective for this research, as the staff turnover in the IS department of the agency has been high. Although the researcher had tried and managed to make contact with a few former staff members involved in the projects, sufficient number of interviewees could not be found. It is essential there are sufficient interviewees to ensure the validity of the study. The researcher decided not to choose it for this research.

Questionnaire

Questionnaire uses a form containing a set of questions that the respondent would choose to answer and return them to the researcher; it would be paper-based or online based. It is normal used for a survey with interest in particular areas and aiming for acquiring statistical data or as a lead for further research in details. It has been widely used in IS research, for example, the widely quoted Standish Group reports were compiled based on questionnaire and the analysis of them.

The advantages of questionnaire related to this research would be:

- 1) The data collection would be in a standard format and easy for analysis
- 2) It would be quick and effective to gather data from a large number of respondents.

However the disadvantages are:

- 1) Research questions regarding the technical issues would be difficult to address in questionnaire.
- 2) The number of the respondents would be hard to predict but a sufficient number of them is crucial to ensure the validity of the research.

Although questionnaire would be an effective way to gather information and to involve more people than interview would. As the target of the research was a selected government agency, using questionnaire still would not produce sufficient data for meaningful analysis. The researcher did not choose it as the research method.

Grounded Theory

Grounded theory is to develop theory that is grounded in data that has been systematically gathered and analysed. It is an inductive, theory discovery methodology, with which the researcher to develop theories from empirical observations or data. The emphasis of this method is continuing interaction between data collection and analysis. It has been widely used in IS research for developing context-based, process-oriented descriptions and explanations of the phenomenon as argued by Myers, Michael D. (1997). It was also strongly recommended by M. R. de Villiers (2005).

The advantages of grounded theory related to this research would be:

- 1) It would produce knowledge that is irrefutable as it only comes out of data on the 'ground'.
- 2) It would lead to in-depth study of the issues as the researcher is fully engaged all the time.
- 3) It would combine with various data collection techniques and data analysis methods.

However the disadvantages are:

- 1) The researcher must be engaged on the 'ground' where the issues are during the research.
- 2) The researcher must use effective methods to collect and analyse data.

Grounded theory would have been a very useful method for answering the questions for this research and had been seriously considered. However, the researcher must be able to present in the actual integration development process and directly related to project staff to do a ground theory research. The researcher could not choose the research method because the integration projects that could be selected for the research all happened in the past.

Case Study

As defined by Robert K. Yin (2002), a case study is an empirical inquiry for investigating a contemporary phenomenon within its real-life context. It was argued to be one of the most common approaches in IS research. The design of case study would be very flexible as it was found in the literature review. Case study would be used for positivistic research as well as for interpretive one. It was argued by Michael D. Myers (1997) the case study research method is particularly well suited for IS research.

The advantages of case study related to this research would be:

- 1) It would allow the research to study the issues within their real life context.
- 2) It would allow the researcher to explore issues and identify answers with almost no constraint.

However the disadvantages are:

- 1) A case study must be carefully designed and rigorously validated.

- 2) The researcher must have in-depth context knowledge and be able to understand the research issues in real life settings.

The researcher decided to choose case study for this research because the advantages would be very beneficial for studying integration projects in a selected organization, and the disadvantages would be mitigated by careful design and rigorous validation of the case study. More reasons for the selection are further elaborated in the following sections.

3.2.3 Selection of Case Study Method

After the evaluation, it was decided that *a multiple case study of interpretive research of qualitative type* would be the most suitable research methodology. Apart from the discussion in the literature review regarding the general advantages of case studies in information system, additional reasons for choosing it for this research are:

- 1) The research project is to investigate integrations of information systems in a government department; hence it is in the information system field. In a paper by Orit Hazzan, Yael Dubinsky, Larisa Eidelman, Victoria Sakhnini, Mariana Teif (2006), they proposed to use qualitative research in computer science education. What they argued in their paper is also applicable to information system research. Target objects of qualitative research approach are usually social phenomena, people, process and technologies. As system integrations within a large government agency are complex issues, and the result of the research is unknown, a qualitative research will help in the two directions at the least. First, qualitative description may help to expand our findings, as qualitative research requires us to collect data from multiple channels, such as documents, observation, interviews and experiments. A qualitative research can open up much more options to collect data and do research than a quantitative research can. Second, a qualitative approach allows us to deepen our findings, as we can choose to concentrate on details of selected process and aspect of the issue. Such approach may lead to in-depth knowledge and understanding of the issue.
- 2) This would be a multiple-case research in the targeted government agency to identify the challenges and solutions of system integrations and applications of

XML. As argued by John Gerring (2007), case study could be defined as intensive study of single or a small number of units for the purpose of understanding a large class of similar units. As opposite of quantitative research or pure science experiment, single or limited number of case study allows in-depth investigation of certain social phenomenon and generation of theories that would be useful in general circumstances. It was even argued that the famous evolution theory was the result of some careful case studies such as sets of fossil on certain species by Darwin.

- 3) In a paper by Bente Anda, Kai Hansen (2006), they used case study on the application of UML in legacy development. They argued that case studies were important if we were to understand the actual practice of software development, and such understanding is an essential prerequisite for the research which to aim for helping the practice. In this research, only by conducting a multi-case study in the real-life setting in the government department, are we able to understand what would be the core issues of system integration and the solutions which might or might not include XML applications.

- 4) Reasons for choosing case study also include consideration of the questions of research, control over the variables and targeted phenomena. As pointed out by Robert K Yin (2002) in his book, case study is particular suitable for finding answers for 'how' and 'why' questions. In our research, most questions fall into these two categories. Case study is suitable for research, which there is little control over the variables, in our research the target is a government agency running on its own course without any control from the researcher. It was also argued that case study prefers real-life and contemporary phenomena over others. The department setting is current and for real, it is exact match to what is described in Yin's theory for an ideal case study target environment. In another paper by Dewayne E. Perry, Susan Elliott Sim, Steve Easterbrook (2006), They argued case study was particularly appropriate when we seek to understand how and why technology is used or not in contemporary settings, and where we have little or no control over the variables. In this research, all the case happened in the past, the researcher has no control on any of the integration variables including the technology of the solution, the systems' platforms, users' requirements and other

setting of the integration or the agency.

- 5) Multiple cases were chosen because the research needed to identify the pattern and the real issues behind the scenes in a numbers of different situations and scenarios. As argued by John Gerring (2007), weak relationship was difficult to observe in a single instance. With multiple cases, the researcher could have a better picture of issues arising from a collection of integration projects. Multiple cases and different stages of a case can also provide comparison for different challenges and their solutions; they are important arguments and foundation for the research findings and conclusions.

- 6) Many theories and papers identified in the literature used the case study method to research on system integration, including critical success factor investigation. For example, in a paper by Wing Lam (2005), the author did a case study in a large finance institute to investigate CSFs for system integration. It first validated a model derived from literature with the case; then the author developed a new CSF model for EAI and used a case from a large finance institute to investigate further. The author argued case study analysis was a well-known approach for exploratory, theory-building research, although it did not always lead to generalization, it would provide rich data and grounding in empirical reality, it was well suitable for investigative research. Many other authors from the review had reflected similar views.

- 7) The researcher has worked for the agency for many years and has in-depth knowledge of the organization, its information systems and issues around them. As the research has been supported by the management of the agency, the researcher was able to used rich source of document and source code for the research, which is normally not available for such research. The context knowledge of the researcher and the availability of various data source would have provided invaluable benefits to the case study.

3.3 Case Study Protocol

The design of case study has mainly followed methods outlined in YIN's '*Case Study Research*' (2002) book.

3.3.1 Case Study Questions

Case study questions were derived from the research questions. They were used for data gathering and analysis. The summary of the answers to the case study questions would lead to answers for the research questions.

The followings were case study questions under the heading of respective research questions:

❖ **What would be the system integrations within a large government agency?**

- 1) What is the integration project?
- 2) What are the goals for the project?
- 3) Why is it important to organization?

❖ **What were the integration solutions?**

- 4) What are the solutions that have been used so far?

❖ **What would be the critical success factors of these integrations?**

- 5) What are the critical success factors for the integration project?

❖ **What would be the methods for resolving these issues?**

- 6) What are issues and challenges faced by the project team?
- 7) What are the major technical issues and solutions for the integration project?

❖ **What would be the XML applications for these integrations?**

- 8) How has XML been involved in past and current system integrations?
- 9) What are the reasons for adopting or not adopting XML in the projects or in the project past?
- 10) How were system design decisions made?
- 11) How successful were XML web service integration solutions when XML was actually used?
- 12) What are the critical success factors for the XML application(s)?

13) What are the reasons for XML web service integration solutions being successful / not successful?

Notes:

The structure of the question was designed to form valid construct for data analysis in later section. Questions for each case would vary slightly to suit individual case.

3.3.2 Case Selection

The research was conducted in a large government agency (called the agency in the rest of the document) with over 3000 staff members. The agency has a large IT department including an application development team. The team had been quite active in system development in the past few years. The researcher have worked for the agency as a software developer for a number of years, he had seen a large number of those development activities. The study by the researcher had been supported by the organization (including special study leave and finance support) for a number of years. The researcher has received explicit permission from the manager of the development team to conduct this research using the projects and the project data repositories.

There are a large number of systems in the agency as it heavily relies on data collection and management in its day-to-day business. There were over a thousand MS Access database system found through out the department in a stock take. However, the target of this research is system integration. It was defined in the literature review, system integration is a process of linking different sub-systems or software applications together functionally, especially for providing a particular result or achieving a particular goal. The selection criteria for case projects were as following:

1) *It must be a nation wide system.*

This would ensure the selected case has relatively large user base and filter out those small systems like MS Access database. As a nation wide system, it would need to be robust in technology and important for the department's business. Hence the result of the research would be more relevant.

- 2) ***It must have interaction of different systems, preferably includes third party system.***

This would ensure the selected case is type of system integration. The involvement of third party software is common and would make the case more relevant.

However, some integration with only internal systems would also be complex and important for the organization.

- 3) ***The development team must be involved.***

They would ensure that there is sufficient research material available to make the case study possible. When the internal development team was involved or responsible for the system, they would have inside knowledge, especially technical knowledge for the project. These knowledge would likely be found within the organization from many sources, such as document, discussion record and system source code.

- 4) ***It should involve some sort of XML elements.***

The research objectives include investigations for finding out what role XML would play for system integration.

The methods and procedures for selecting the cases include:

- 1) ***To search in the agency's document store***

The agency has a central document store for all its documents, including those for IT systems. It was set up like a folder structure. The documents for a particular IT system are stored within one folder for the system. There are large collections of these IT systems. Most documents are open to the agency's staff except those containing personnel or sensitive data. As staff member of the agency, the researcher has access rights to these system related documents and had searched them for possible cases for the research.

- 2) ***To search in the agency's source code repository***

Most of the source code for the agency's IT applications are stored in the source code repository, including the Microsoft SourceSafe database and Microsoft Team Foundation Server. As a member of the development team, the researcher can access and look into source code. The researcher had done a thorough search in the repositories for selecting the cases.

3) ***To search on the agency's intranet***

Most of the nation wide systems of the agency are placed on the agency's intranet site. They are available to all staff members with necessary security control implemented within individual system (including backend database access). This is the easiest way to access current systems and can be used for some testing and validation for findings that had been acquired from other sources. However, the intranet site only has current systems; all historical system must be searched for in other sources. The researcher had used it mostly for testing and validation of current systems.

4) ***To discuss with current and former colleagues in the development team***

Discussion with people involved with the system development would help to find out where to get more information or to do initial evaluation on the suitability of the system for the study. The researcher had talked to a number of staff members for selecting the cases.

By going through all methods and use criteria outlined above, the researcher had identified and selected the following cases:

1) ***Geospatial Information System (GIS) Integration***

It was an integration of GIS system with a number of internal business systems.

2) ***Document Management (DM) Integration***

It was an integration of DM system with a number of internal business systems.

3) ***Office & Person Data Sharing***

It was a set of projects and attempts for the purpose of sharing office and person data among all internal systems.

4) ***Corporate Directory***

It was a project attempting to centralize processes related to a number of internal and external systems.

There were also a couple of other projects the researcher had considered but decided not to select in the end, they were:

1) **The Asset Management Information System (AMIS) SAP Implementation**

It was a large customisation and implementation of SAP ERP system. However, all work was done by external consultants, the internal IT team has little involvement and little information about the system is available.

2) The agency's Intranet

The intranet is a large system integrated with a few other systems, but the integrations are very shallow including mainly hyperlinks. There are not much integration elements or XML.

In summary, this is a multi-case study research. As argued in the methodology selection in the previous section, a multi-case approach would better help the research to identify elements and their relationship for integration system development, including issues, solutions and the XML element. All selected cases had happened in the past, some of them would have started from nearly ten years ago. The collection of the above cases would be a good representative of system integration in the agency. The total number of four cases (some individual case may contain multiple systems or attempts) would have provided enough material and evidence for valid research conclusions. It was also the result of the availability of the suitable material for the researcher to handle for the purpose of this research.

3.3.3 Data Collection

Since this is a multi-case study on four integration projects all happened in the past (some for a long period of time) within the agency, the data collection methods were designed accordingly. As a staff member and software developer for the agency, the researcher not only has access to a large quantity of material including documents and source code, he also has the background knowledge required to understand and analyse them. Hence, the data collection was designed to take the full advantages in such settings.

Interviews

This method was tried but it was only used rarely in this research. The reasons were:

- 1) Interview is an effective method for analysing human behaviours and management issues, but they are not the main interest of this research.
- 2) Interviewees with technical background were crucial for the case study. One of the main interests of this research is in technical issues; interviewees need to be with strong technical background to answer all case questions.

- 3) A large number of interviews would be required. In order to get a balanced and comprehensive view on a case or an issue, many interviews would be required. The workload of doing interviews for all four cases would make such approach neither suitable nor effective for the research. There would be multiple phases (small projects by them self) within some of the cases, which would multiply the number of interview that would be required and make this method even more unrealistic.
- 4) Interviewees are hardly available. Most people involved in the projects have left the agency when the research was conducted. Although many attempts had been made to identify potential interviewees, only a small number of interviews had been conducted in the end.
- 5) In this research, other forms of data source were more reliable and readily available, including project documents, source code and the systems on the current network.

Initially, the researcher had prepared and conducted a few interviews including one with a consultant in another government agency another one with the former GIS DBA of the agency. The interview with the consultant did provide some useful data; however, since the focus of the research was changed to the selected cases within the agency, the data become unused. The former GIS DBA was not be able to answer any of the XML related questions, as he has not been exposed it. In many other occasions, when prospective interviewees were contacted initially, they were happy to agree being interviewed but put it off later after receiving the interview (case study) questions, many of them would require technical expertise to answer. In summary, interviews would not be the main data source for this research; but because they are important for strengthening the triangulation of the study, a limited number of interviews were conducted when suitable interviewees were available.

Documentation

Documents were the primary source for this research and they were usually collected in the first step of data collection. There are a large amount of document for all the projects, including the business case, user requirements, system architecture and design and functional specifications. All these files can be collected from various repositories, but it would be unnecessary and impossible to go through all available documents. For

the case study, the objectives for reading these documents and information that should be gathered from them are:

- 1) To answer the case study questions.
- 2) To collect document artifacts.
- 3) To collect evidence for any opinion.

The main reason for using documentation as the primary data source for this research project is because of the nature the integration projects included in this case studies. Most of them were long-running projects with multiple stages. Many of the staff members involved in the projects had left the organization but documentation had stayed. Some of the document would be dated back as long as 15 years ago, but only the recent ones were closely studied in most cases as the information from them would be more up-to-date and relevant. By going through documentations in various repositories, it was an effective way for gathering data.

Other reasons for using documentation are:

- 1) Documentation would be one of the main data source for any case study, according to Yin (2002), the advantages of documentations included:
 - i. stable - can be retrieved repeatedly
 - ii. unobtrusive - not created as a result of the case study
 - iii. exact - contains exact names, references and details of an event
 - iv. broad coverage - long span of time, many events, and many settings
- 2) Documentation can provide multiple sources for any opinion, which is one of the strength of this case study.
- 3) Documentation can be cross-referenced and validated, which would improve the reliability of the research.

Documentation also has weakness, some common ones are:

- 1) It can be biased, if the collection is incomplete or the author is biased or ill-informed.
- 2) It can be hard to access, if it is confidential or deliberately blocked from the public.

However, these weaknesses would not be a particular problem for this research.

- 1) The first weakness would also be applicable to other form of data source including interview. It would more likely be an issue for interviews as people often change their opinion during time, or there would be misunderstanding between the researcher and the interviewee.
- 2) The second weakness would not be a problem in the research, because
 - i. The researcher has all the necessary security access to those documents.
 - ii. The researcher had permission from the management to do the research.
 - iii. The research was focused mainly on technical issues that would involve little sensitive data for commercial or management purpose.
 - iv. The researcher would not need to use any confidential data and had excluded them as soon as they were identified.

Source Code

Source code is very important for this research and a lot of information had been discovered from inspections of source code by the researcher. Source code is important supplement for general documentations, especially for understanding system design and functionality, as such information may not be found documented in any other document format. Source code and database schemas are ‘physical’ components of the system; they would reveal the ‘true’ nature of the system more than any other form of data source would. Source code would not be normally available or used for reasons including the following:

- 1) They can be copy right protected or sensitive Intellectual Property (IP) issues can be involved.
- 2) They require specialist expertise to understand, such as .Net C# and SQL knowledge.

However, they were not a problem for this research, because:

- 1) The researcher works within the agency, he had received support from the management and written permission for doing the research. There was no any commercial motivation involved in this research.
- 2) The researcher has over 6 years experience in C# and 12 years in SQL. He has been a Microsoft certified Solution Developer for over 9 years. The researcher would not expect any difficulties for understanding the source code.

Source code is very valuable data source for this research. They had been used:

- 1) To identify change history of the system that may not be document, such as original author and date time stamp of the system.
- 2) To understand the database and system design, some of design information could only be found in developers' comment embedded in the source code.
- 3) To uncover other details or facts which are not found in other documentation, such as the style and quality of the system.
- 4) To confirm and cross-check information in other form of documents, such as design proposed in document and the actual one in production.
- 5) To re-create system architecture diagrams when there was no one found in the design documents.

Discussion

Apart from interviews, there were a few informal discussions between the researcher and relevant staff in the department's IT team. Discussions would be useful in many ways. The benefits of discussions were:

- 1) They could provide supplement information, such as background for a design decision, which can be hard to find in the documents.
- 2) They could provide good starting points for other data gathering processes, such as documentation search and system testing.
- 3) They could provide overall pictures of the integration projects.

System Testing

Some testing had been conducted when the target systems were available. Most of system testing was used for verifying system features and design. For those integration systems with different design approaches, system testing has provided basic data for comparison, including system features and performance. They would provide import information from basic fact-findings to data analysis.

3.3.4 Data Analysis

The core objective of the analysis is to identify relations from research findings. As argued by John Gerring (2007), case study analysis usually is focused on the identification of casual relations, not on the elimination of possible causes. The

analysis of the data was organized around the research questions. It was done first for each individual case and then for the research as a whole.

Analysis Principles for the Case Study

To ensure the validity of the research, various analysis techniques considered and applied where is possible in this research would included:

1) Triangulation

As found in the literature review, triangulation for IS research is about multiple methods, multiple data source, multiple analysts and multiple analysis perspectives. For case study, it was described as using multiple data sources and/or types of evidence converging on a 'fact' by Yin (2002). It was argued by Lynne Johnstone (2007) that it would help to resolve the questions about contradictory evidence and the equality of data in studies involving multiple types and/or sources of data. Triangulation is not about getting the same 'fact' or seeking the same result; it is for understanding different ways of seeing data and inconsistencies of different kind of data which would lead to deeper insight of relationship and phenomenon. In summary, triangulation would help to reduce the impact of potential biases and improve the reliability of the research.

Triangulation would be applied in this research in the following steps:

- i. For each case, in answering the case study questions, data were gathered from multiple data source including document, source code, interview and system testing if available. Data from various sources would be gathered and validated against each other. Detail findings and initial analysis for each case were documented in the appendix section.
- ii. In the summary of all the cases (section 5.1), answer to each of the research question would be gathered as summary from the individual case. The principle of triangulation would be applied to seek certain viewpoint to be reflected in multiple cases or to explore reasons of any inconsistency.

2) Chain of Evidence

A chain of evidence would help reader to understand how the results have been achieved, and hence improve the conformability of the case. As argued by John Gerring (2007) and Robert K. Yin (2002), process tracing and forming chain of evidence are import data analysis techniques for case study. The construct of the

case study questions, especially those for XML application (question 8 - 13) were organized for forming chain of evidence.

3) Root Cause Analysis

It is a process of finding the most important reasons for an issue, such as the success or failure of a project, by repeated processes of verifying and correcting the previous finding. As it was used by Wing Lam (2005) to identify critical success factor (CSF) for enterprise application integration (EAI), root cause analysis would be used for answering certain research questions, such as for identifying CSF for the integration project, the core technical issues and solutions. The technique was used for finding answers for case study question 5 – 7 and also in analysis section 5.2.

4) Counterfactual Comparison

The technique was applied in answering study question 8 - 13 in case 4. As argued by John Gerring (2007), when there is no other practical method available, counterfactual comparison is plausible; otherwise the research will be constrained on those cases with controllable variables. As all cases were from the past and no controllable variables for any of them, answers of certain case questions could use counterfactual comparison for cases when different integration techniques were used.

5) Pattern Matching

As argued by Winston Tellis (1997), replicating pattern-matching could strengthen multiple cases results and could increase the robustness of the theory. The method had been used in the summary of all the cases and in the conclusion for this research.

Validity Concerns

The researcher had seriously considered validity of the study according to the guideline identified in the literature review. Details and explanation for the validity itself would be found in the literature review section. The followings are the validity consideration for this research.

Internal Validity

This is related to issues that may affect the causal relationship between findings (& data analysis) and outcome. The threats would be:

- 1) All projects were in the past and the circumstances would have changed completely.
- 2) The technology and project environment have change since the project happened.
- 3) The selected projects would be bias or not suitable for the research topic.
- 4) The collected data would be bias or not complete.
- 5) Most people involved in the project had left and would not available for interview.
- 6) There would be hardly any valid testing in for any opinion.

External Validity:

It is about generalization of study result from the samples included in the study using inductive logic. The threats would be:

- 1) Each project in the agency would be unique.
- 2) The data for the case study would be inaccurate.

Construct Validity

This concerns the logical confidence one can have in whether the variables in a study are valid measures of the corresponding constructs in the theory being tested. The threats would be:

- 1) The collected data would be bias or not complete.
- 2) The answers to the research questions would not lead to valid conclusions.

Conclusion Validity:

This criterion concerns the validity of statistical conclusions drawn about the relationships among the variables investigated in the study. The threats would be:

- 1) There would be not statistics evidence in any case.
- 2) The total number of four cases would not be used as statistics evidence.

However, conclusion validity would be important if the conclusion is to be generalized, which was not the focus of this research. The concern of conclusion validity for this research was minor.

In summary, to ensure the validity of the research, all possible threats should be identified and then actions to be taken to neutralized them. Some of the threats would be common for multiple validities, so it would be more important to address them in

the research. Remedy actions are described in the following quality assurance section. Because this is an interpretive research, the conclusion would not need to be generalized; therefore, the threats to the conclusion validity were not specifically addressed.

Quality Assurance

The following is a table for quality test designed by Yin (2002) and the action to be taken in this case study research.

Test	Tactic	Actions applied in this research
Construct Validity	Use multiple source of evidence	The data was collected from documents, source code, system testing and interviews.
	Establish chain of evidence	The construct of research questions forms the top level of the chain of evidence. With the answer for each of those questions, evidences would be traced backward and forward in the source code, design document and the system itself.
	Have key informants review draft case study report	The report had been reviewed by a system analysis and a former developer involved.
Internal Validity	Do pattern-matching	Pattern matching was applied on multiple data source. For example in case 4, the pattern was identified when multiple sources reflected on the causes for the project failure.
	Do explanation building	In answering the research questions, explanations were explored from data sources.

	Select case with set criteria	All the cases selected were integration projects involving multiple systems to different degrees.
External Validity	Use replication logic in multiple-case studies	For all the four cases, the researcher following the same logic and use the same set of questions.
Reliability	Use case study protocol	The protocol was designed before the research and has been refined during the research.
	Develop case study data base	All collection of documents and interview records have been created and stored accordingly. They have been constantly referred to during the research.

Table 2. 9 Quality Assurance Measures

Review of Draft Report

As part of the validating procedure, the draft report had been reviewed by a number of external IT professionals for peer debriefing of the emerging insights in the research area. As argued by Yin (2002), review of the draft report was related to the overall quality of the study, it was to corroborate the essential facts and evidence presented in the report. When a reviewer disagrees with the conclusions, she/he may not disagree with the actual fact of the case. It often becomes an opportunity for searching for further evidence.

The report has been reviewed by:

- 1) a senior consultant from a IT consultancy firm
- 2) a system architect from a major IT company
- 3) a senior system analyst in another government agency
- 4) an system architect in another government agency
- 5) a developer from a major IT company

Feedbacks from the reviews were very useful and some of them were found crucial for the quality assurance of the report. They were ranged from comments of the thesis

structure to opinions of the final conclusion. Most of them have been incorporated into the main body of the thesis.

Principles for Interpretive Research

As this is an interpretive research, the study had followed the principles proposed by Klein and Myers (1999). The principles were reviewed and explained in the literature review, the title of the principles and actions following them in the study are summarized as following:

1) ***The fundamental principle of the hermeneutic circle***

The researcher has been through many circles of the investigation within the study. First, data was collected and the analysed for each of the questions for a case. As some questions were built on the others, repeated circles have been through for answering all the questions. Second, in the end of each case, the research did a summary of the case by going through the case in another circle. Third, the researcher has repeated the circles for all the four cases in the study one after another. At last, the researcher has done another set of circles based on the summarized data of all the cases.

2) ***The principle of contextualisation***

In this study, the researcher had work for the agency for many years, it has help a lot in understanding in the background of the research. During the research, the researcher has done extra search and data collection to ensure he could see how the situation under investigation emerged.

3) ***The principle of interaction between the researchers and the subjects***

The researcher had good understanding the agency's organization structure and knowledge of the people involved in the projects. Since the study mainly relied on the documents left behind from the projects, the concern of this principle would be lessened. However, the researcher had been curious and had avoided any comments and statement without supporting evidences.

4) ***The principle of abstraction and generalization***

In the data analysis, after the researcher had summarized the facts, he had identified and extracted those essential data, and then applied with explanation and theories.

5) ***The principle of dialogical reasoning***

As the researcher had worked as a developer for the agency, he had paid extra

attentions for keeping himself as an neutral researcher during the investigation. Since all projects had happened in the past, the research would not have any impact on their outcome anyway. To ensure the researcher's own ideas from before the study would not influence the investigation, the research had based the study on solid evidence only, such as source code written for the system in the past.

6) *The principle of multiple interpretations*

When the researcher summarized the case data, he would look for evidence that supported different opinions. Since the study was focused more on technical area, the concern of this principle would be lessened.

7) *The principle of suspicion*

The researcher has been aware people involved in the projects would have their own point of view or limitation of knowledge. Through the actions taken under the previous principles, the researcher had reduced the risk from possible 'biases' and systematic 'distortions' from the participants.

3.3.5 Report Format

The rest of the report would in the following format:

- In chapter 4, the summary for each case would be presented. The bulk of data collection and preliminary analysis for each case was done one by one. Most data collected for individual case is placed in the appendix for reference only.
- In chapter 5, summary of the findings for all cases and the bulk of analysis for the research would be present. Analysis and answers for the questions raised from the literature review would also be presented.
- In chapter 6, the research conclusions would be presented. Further discussions including the integration development models would also be included.

Chapter 4 Case Summary

4.1 Case 1: GIS Integration

4.1.1 Project Overview

Since early 2001, the agency has established its GIS platform based on ESRI software products, including ArcSDE, ArcEditor, ArcReader, ArcIMS and etc. However, they are expensive software and only a small number of licenses have been purchased. The agency has a number of other core business application systems, including Land Register, Permissions and Bioweb, they are in-house and Microsoft based thin client applications. Land Register stores and maintains records of all government own land passed to the agency for administration. Permissions system is used for processing permits for all kinds of activities that happen on the land. Bioweb is a collection of applications for keeping records of species and observations around New Zealand. All these systems require integration with GIS with requirements including showing business data on the maps and running spatial queries on the maps using selection criteria applicable to the business systems.

The integration had been a long running and multi-phases project(s). In the earliest phase before 2003, the integration was established by simply importing business data into the central GIS database by using data import stored procedures. Users used the specialist desktop client application licensed from ESRI to work with all GIS and business data. There were limited sets of business data and they were disconnected from their business repository. The main issue of such approach was that the GIS database was getting bigger and bigger as duplicated data had been created. In a later phase between 2002 and 2003, a browser-based interface (DOCgis) was developed by customising an off-shelf ESRI map viewer; business data were linked to the central GIS database by using views through linked tables. DOCgis works on the combined sets of data, based on the core GIS database and other business data through views established in the GIS database. More and more business data could then be added to the data sets, which would allow the integration to include more and more applications. Apart from issues associated with duplicated data in the GIS databases, there are other issues such as the integration is strongly tied with the business systems through linked tables and views. There have been many incidences when changes in the business databases had caused errors for the GIS database and DOCgis.

In 2005, a new GIS integration strategy had been adopted for adopting the latest .Net and XML web service technologies to develop a new GIS integration solution. It works on the core GIS database through the ArcIMS map service and a set of web services for retrieving data from various business systems. The ArcIMS map service is responsible for providing map images generated from the core GIS database; all other business data are retrieved through dedicated web services. In addition, the new solution works on an XML configuration file; it also includes new functional integration features such as hyperlinks between various parts of other business systems and the maps. So far the new integration (GIS Viewer) has been fully integrated with Permissions and Lang Register systems.

4.1.2 Consideration for Research Questions

GIS systems are becoming more and more popular with large organizations, especially with government agencies. In fact, the agency had a specialist GIS team responsible for coordinating and developing GIS application for the department. GIS applications are also becoming more and more feasible and affordable as the technology becomes more accessible. For example, the latest release of Microsoft SQL Server 2008 has all the basic GIS features included, traditionally expensive license fee for specialist GIS software may no longer be required.

The agency is responsible for the administration of about two third of the New Zealand land mass and most of the coastline and sea territory. Fully functional GIS systems are essential tools for the agency to carry out its missions. A successful GIS systems in the agency must be able to provide users with complete and accurate information, which may include not only just GIS data such as maps but must also include other business data such as those from Land Register, Permissions and Bioweb.

The challenges during the development of GIS integration would be common for large government agencies. The challenges were highly complex as they involved specialist software such as ESRI GIS package, a large number of business systems such as Permissions, and large amount of data from various sources. They are important and would have crucial impact on the department business, for which the specialist GIS team is tasked. By investigating into the GIS integration in the agency, it would

generate highly relevant results for common system integrations for the agency and other government agencies.

In the GIS integration development for the agency, XML has only been used in its later stage since 2005. The case study would be an excellent example for investigating and comparing different integration strategies (with and without XML). XML applications have been extensively used in the integration development since 2005. For example, XML was used in the configuration file, for the web service interface and GIS queries. The case study would certainly provide answers for the research question.

XML has been adopted since 2005 for the new GIS integration (GIS Viewer). The case study would investigate why and how decisions were made to choose XML, including business and technical rationalization. The investigation into the decision and selection process would provide inside processes of IT department of a government agency.

The deliverables of GIS integration projects have become essential tools used by the department users. Since they have been used in a day-to-day base by such large user base, it would be important to find out how useful and successful the new XML-based solution is.

Since most of the GIS integration system has been developed in-house in the agency, the case study would have a rich source of business and technical knowledge of the system. There are a large number of documents, system design and even source code for the GIS integration system. All these knowledge would help the investigator to track down and identify reasons for the solution being successful or not.

4.1.3 Summary of Case Findings

(Please refer to the appendix section for detail of findings and preliminary analysis)

- 1) GIS integration would be regarded as a combination of data, function and interface integration. It combined data of GIS (Maps) and business (Permissions) data. It could generate reports across GIS database and multiple business systems. It also provides a central user interface for interacting with those systems.
- 2) GIS integrations are common and important for large government agencies such as the agency, as it was argued in the business case and requirements document. GIS

systems need to combine data from other business systems to provide rich information including maps and reports for business purposes. Integration requirements usually include spatial queries on multiple systems and linkages among them, such as those specified in the Permissions GIS integration requirements.

- 3) The project has been a long running one over many years including two main streams of DOCgis and GIS Viewer respectively. There were many management and technology factors affecting the development, including the arising of new business requirements, availability of tools and other development resource (developers). They were the main reasons for the different approaches of the two GIS integration projects.
- 4) The early attempts of DOCgis trying to import and link data entirely from the backend databases as it was indicated in the NEGIS architecture diagram. It was a 'quick and easy' approach for simple integration features by limiting the scope within the backend databases, but it had many drawbacks, such as redundant data and maintenance difficulties. Most problems were caused by such tight coupling approach within low-level database objects such as linked tables and views. Such approach would also only be feasible for single platform database server; any cross platform databases would make it very difficult if it is not impossible at all.
- 5) There were other major issues of the old DOCgis such as the lack of spatial query ability and functional integration ability. They were the main reasons for the development of the new GIS Viewer when such requirement had arisen from the business. The new integration functions were essential enhancements for the business systems as they were described in the documents, including the Permissions II requirements, and Permissions GIS requirements.
- 6) The new GIS integration development was designed for fulfilling a new comprehensive set of integration requirements. It had been through rigid development processes including software evaluation, proof of concept and prototyping. The new design focused on integration of multiple systems through purpose built interfaces such as XML web service, as it was indicated in its system architecture and verified in the source code. The new approach was successful one as the project was delivered on time and within budget. It has become one of the major systems for the agency, as the usage statistics of the web server had indicated; it has hundreds of regular users of a similar number for Land Register. It

is also flexible and easy to maintain as new business systems such as Land register had been integrated with it with minimum cost.

- 7) The new GIS integration development has taken advantages of XML applications in many areas. Component using XML are the core parts for the new GIS Viewer project, including the new user interface and the XML configuration for business system integrations. XML web services are used for data integration to achieve loose coupling and easy maintenance. XML was a critical factor for the overall success of the GIS Viewer project.

There are some special conditions or limitations in the GIS integration case. They are:

- 1) All sub systems are based on Microsoft SQL server databases, linkage among databases are easy and without any restrictions. Without such database platform, the first integration DOCgis could not be able to link database objects through the backend server, or it will be much harder to achieve data integration only using database objects.
- 2) For the new integration of GIS Viewer, development team were allowed to choose technologies from all the latest and greatest ones including XML, which is not usually the case for system development in large organizations. As from discussion with the analyst who overseen the development of Permissions and Land Register, many of the department systems, including Permissions and Land Register, were developed under strict time frame with pre-defined architecture. XML might not be used at all for GIS Viewer if existing software has to be reused or other third party component was chosen.

Current status of GIS integration:

- 1) The old DOCgis is still in use as one of the GIS integration interfaces. Other GIS desktop software such as ArcReader and ArcEditor are also be used by a few GIS specialist in the agency. Problems with its database integration pop up from time to time which regarded by some as system 'hiccups', as those connection are hard to find and easy to break when the database are changed. GIS databases are the major payload on the database server, as it has many regular and intensive processes for transferring data from business systems.
- 2) The GIS viewer is now a major business system like Permissions and Land Register. It works well except when the GIS backend server has its 'hiccups', as

the GIS Viewer relies on the map service which depending on the backend GIS databases. GIS viewer has not replaced the DOCgis for a few reasons. As stated in the Permissions GIS integration design document, GIS Viewer was designed for enabling the new Permissions GIS integration, not replacing DOCgis, otherwise power conflict would have arisen and new GIS Viewer might not be allowed to go ahead. As it was indicated in the requirement document and discussions with people involved, the former GIS manager was very adamant in upholding the DOCgis as part of the NEGIS framework and only support a limited and specific GIS integration development for Permissions and Land Register systems.

- 3) There had been ambitious plans for re-develop the entire GIS database to encompass entire sub-systems, such as Land Register; new business case had been created and documented. However, the new system requirements have never been clarified; and the proposal did not receive enough support. Since the CIO and GIS manager for the initial proposal have both left the agency, there is no plan or budget allocated for it at the moment. Would the project eventually happens, it would be a total different integration approach, most likely to be an ERP approach with GIS capability as the business case had suggested.

4.2 Case 2: DM Integration

4.2.1 Project Overview

The agency purchased and installed a document management (DM) product from a software company from Australia in 2000. The agency has been using DM since for storing and managing all business related documents, including all kinds of files, e.g. MS Word, Excel and PowerPoint documents. DM system has similar structures as normal file folders, but it provides a full set of document management features such as GUI document security management, document versioning and quick access shortcut including URL links to documents.

Because DM is the central repository of all business documents for the agency, integration requirements have arisen since early stage because users wanted to link records in business systems such as Permissions and Land Register to documents stored in DM. Access to DM using document shortcut links from within business applications are also required. As DM is a third party software product from an

external provider, looking into and modifying its source code would not be possible. The integration of DM with other in-house business applications has encountered many challenges. Various approaches have been adopted for fulfilling integration requirements in different stages.

4.2.2 Consideration for Research Questions

Document management systems are common in large organizations, such as the agency. They are essential tools for manage large amount of business documents. There are large selections of document management software in the IT industry, as they can be found in popular web sites, such as Wikipedia (<http://en.wikipedia.org>). Many document management systems attempt to integrate document management directly into other applications. Such integration is commonly available for office suites and e-mail or collaboration/groupware software.

DM is one of the essential tools that the agency's users use in a day-to-day base. For any organization that deals with a lot of documents, some form of document management is necessary. It is inevitable that the document management system in the agency to interact with other business systems, such as to share data and to maintain connections. Integration of DM and other business systems has long been necessary and desirable in the agency. The investigation of the case will help to find out and reveal challenges and solutions for such system integrations in a NZ government agency.

As DM is a third party product without any access to its source code, the integration of DM with other department business applications has created a lot of challenges. These challenges can be common for integration for existing systems, which were developed separately or supplied by different vendors. For most third party software, no inside design or source code may be available for any integration projects. The challenges and experiences arising from the DM integration in the agency would not only benefit the agency, but would be very useful for other government agencies, as such integrations were found quite common. As the case document revealed, the Public Trust had a very similar integration using DM at the time the agency was developing its own one.

The DM integration project has been through a few stages and it was developed within the agency entirely. The XML web service was designed and developed to overcome issues that caused by incompatible software system. The case study would reveal the full details of these issues and provide explanations why and how XML web service became involved. Most XML applications had been introduced as part of the XML web service application for the DM integration in the later stages of the project. The case study will compare the approaches and results of using the only DM library components without XML and the approach of adopting the XML web service.

There might be good reasons for both adopting and not adopting XML in the DM integration project depending on the business requirements and application designs, i.e. for simple document creation on the client side, no XML would be necessary, but for server queries, adoption of XML web service would be a better solution. The project had various business requirements and was a good example for investigating the research questions.

The DM integration project has developed a few new components, including the COM+ applications and XML web service. The agency users in a day-to-day base use those DM integration functions as part of the business application for many years now. The case study would find out how successful the solutions are by discussing with users, investigating the databases and performing testing on the systems.

Since the DM integration project has been developed within the agency entirely, the case study has a rich source of knowledge for answering the research question. There are a large number of documents, system design and even source code for the DM integration application. All these will help the research to identify reasons for the solution being successful or not.

4.2.3 Summary of Case Findings

(Please refer to the appendix section for detail of findings and preliminary analysis)

- 1) DM integration would be regarded as integration of common functions and interfaces. Document management systems would be common for large

organizations including government agencies, so would be their integrations with other business systems. The integration requirements arose very early in 2002 soon after the DM implementation in the agency and at the same time Permissions system was in development. It was an important part of the Permissions system right from the beginning, as the user requirements document and Permissions user guide have shown. As the trainer commented, users were impressed with the integration feature. A large amount of document relevant data has been created in the database from the integration.

- 2) Integration with third party software would present extra challenges and require more skills and effort. Knowledge of the product only could be accumulated after large amount of effort had been spent. It had been a long learning curve from the initial phase when 'quick-and-dirty' design was adopted because of the lack of development time, to the current status after many rounds of software upgrade and changes. The integration system had been through a lot of changes as reflected in the system design and source code.
- 3) The DM integration project was also a long running one consisting multiple phases. Not every part of the integration requires XML, but more and more XML applications had been adopted as the DM software and other business have evolved over years. As using XML web service is part of the best practice for passing data between application, and XML web service is deigned for machine-to-machine interaction including integration. XML web service was the key for the integration to overcome the incompatible .Net version problem and improve the flexibility of the integration component.
- 4) The XML web service would be one of the critical success factors for the project. The complexity of the integration and extra work had been caused by different version of software, especially the client side DM installations. It had illustrated that without flexibility such as those from XML web service, it can be difficult for maintaining the integration over a long period of time and for different systems. On the other hand, the use of XML web service for resolving the different .Net version problem and the subsequent quick integration with Land Register had illustrated the benefits of XML. The XML web service would be well placed for more future integration, as it was stated in the design document.

Special characteristics of DM integration:

- 1) For old fashion client-server systems like DM, which is also a Windows COM based applications, XML could hardly be used. On the client side, its library DME.exe only takes pre-defined parameters; they are all of simple data type. No complex data type like XML document would be used. Using XML web service to execute certain task such as creating document would be impossible, as the software was designed on some strict rules, such as only accepts the Windows network credential from the current thread, as explained by the vendor in the teleconference session.
- 2) Integration with third party software such as DM can be very challenging, as there may be little support or documentations. DM is one of the document management software packages that do not have a large client base for sharing issues and solutions; neither does the vendor provide technical support promptly. The software vendor only has limited capacity, as from the email document, one of the reasons for being late to get back to the agency's development team was there was no technical staff available at the time.

Current status of DM integration:

- 1) The DM integration has become core part of the business systems. Documents describing potential integration requirements for other business systems including AMIS were found during the research.
- 2) The XML web service for communicating to the DM server would be enhanced or extended for better performance or functionality. It would be modified to directly query the DM backend database, as the current query through the library component would be very slow as it was noticed in testing. The web service would also be extended to include functions for creating new document, so the complicated client side process will be avoided.

4.3 Case 3: Office & Person Data Sharing

4.3.1 Project Overview

The department's office, its structure and staff personal data are included as part of the basic set of data in most of the agency's business system. In some systems, these data are used to control the access and other security mechanism for the applications. In the

past, each application had its own set of tables for these data in their individual database. From long time ago, there has been a topic and potential project for integration and centralization of all office and person data in all departments' systems.

In 2002, when a new system (Permissions) was in development, the office and person data integration topic was raised again and some actual implementation was done for the first time. From then onward, there have been a series of effort and attempt to try to get all internal systems to use a common set of office and person data shared by all applications. In one of the attempts in 2006, an XML web service has been developed for providing common office and person data. Somehow there was a lack of adoption of the new web service. Since then, there has been another round of effort to tackle the issue.

4.3.2 Consideration for Research Questions

Office and person are basic set of data for any large organization. They are essential data in many business applications, especially for access and security functions. There are obvious benefits for maintaining a commonly shared copy of office and person data, and integrated them with all business applications. For example, it can ensure consistency and integrity of these office and person data across all business application; it can reduce the maintenance cost of these data. It is important for the agency and any other organizations to use and maintain those organizational data such as office and person, in a consistent and easy manner, ideally with system integration or data sharing.

Sharing common data in a large organization is a common IS integration requirement. The project for integrating office and person in the agency has been a long running one and still without much success. The case will be a good example for trying to identify issues and solutions for system integration in large organizations.

Office and person data are basic data for any business systems in government agencies. The sharing and integration of those data would be important and also common for the organizations. As the project is still an ongoing concern for the agency, the

investigation of the case will not only answers the research question, but will also help providing future directions for the agency.

As XML is general regarded as one of the best format for data sharing, it will be an excellent example for this project to consider using XML. The office person data sharing was first established as direct linking in backend database tables, then XML web service as been developed. The case study will directly address the research question and find out how XML can help to achieve the integration goals. XML has been involved since the office web service application was built. But because of some issues such as slow performance, the service has not been very useful at all. The case study will reveal the full details of these issues and explore possible solution for the future.

There were both good reasons for adopting and not adopting XML web service in office person data sharing. One of the biggest concerns is the impact on application performance by using XML. More work is still required for the XML approach to work. Although the current XML web service may not be seen as very successful, but it still seems to be a step forward and points us to some future directions. The case study is an excellent example to investigate the XML application in data sharing and integration, and provide answers to the research question.

4.3.3 Summary of Case Findings

(Please refer to the appendix section for detail of findings and preliminary analysis)

- 1) Office and person data sharing would be regarded as a data integration project in general. Organization data like office and person are essential data for many business systems. One of the PerOrg project documents found in during the research had identified over a dozen of the agency's systems as shown in the architecture diagram in the findings section. Integrations of office and person data among business systems are important for the department which was argued in many of the documents for many reasons including data consistence, security and usability listed in the findings. As argued by Janusz R. Getta (2006), one of the central problems in the development of enterprise information systems is an efficient integration of data across sub-multiple systems.

- 2) The office & person integration project has been a long running one without much success. The earliest project document found was in 1999, and the latest DLL component was just deployed in production. There were many attempts and even solutions, but none of them would have achieved the ultimate goal of sharing the office and person data across all business systems. The main issues identified from the research include no solid business requirements were specified and suitable software platform was not identified and used. As argued in the Standish Group (2004) report, clear and basic requirement is one of the top critical success factors for IT projects.
- 3) The attempt of using XML web service was not successful, even the original developer has developed a new DLL component to replace them as the research found. The main reason was that DLL library is much easier to use as argued by the developer. The researcher had found the design of the web service was very narrowly focused, such as for providing data for dropdown lists. As argued by Zhengming Tang, Yana Kadiyska, Hao Li, Dan Suciu, James F. Brinkley (2003) in their SilkRoute framework, XML data can be dynamically and efficiently generated, and such process can be applied to any database and its data can be mapped to any number of DTD/XML schemas. Simple XML web service like the one developed in the agency for office and person data could not be the solution for the integration project. Many other business systems like Permissions requires a lot more than what the XML web service would provide.

Special characteristics of the office integration:

- 1) It was argued in one of the documents that the main reason for having multiple data sources for office and person was due to the lack of clear ownership of those data, and it was also the main reason why any of the integration projects was not properly funded. As argued in the Standish Group (2004) report, support of the executive management was the most important critical success factor, the research found no evidence of management support of the project but many unfinished and unfounded requirements and other project documents. Most work completed was done within the scope of individual systems, such as the common database for Permissions and the XML web service and DLL component for Bioweb.
- 2) XML web service with only simple functions would not be very useful. The research found advanced XML techniques, such as Xquery is not available in the

development tool set for the department or commonly understood by the department's development team. Most the existing XML web services in the department are for data access to the backend databases including getting data for UI controls, which the office and person XML web service was designed for.

Current status for office & person data sharing:

- 1) Since all business systems involved in the integration are Microsoft SQL server based, the integration can be achievable using a central database, which the research found two business systems (Permissions and Cites) are currently using. The development DBA has also put in some new effort in creating the new 'commonDB' database in mid 2008 and facilitating the data sharing using the centralized database.
- 2) Since the new DLL library was created for replacing the XML web service, there were some discussion of the Pros and Cons of the two approaches. It is still to be found whether the DLL component is truly better than the XML web service. Advanced XML technologies such as Xquery would be also adopted in future if the development team has such capacity.

4.4 Case 4: Corporate Directory

4.4.1 Project Overview

In 2006, the department initiated an IT project called Corporate Directory. The objective of the project was to provide a single approach to managing the information and processes that relate to a person starting work at the agency, changing their details or IT resource requirements whilst here, and managing information about their departure. It was an ambitious project involving integration with a few existing IT systems. They were Active Directory on the agency's Microsoft Windows network, the agency's intranet database, the agency's HR payroll system and the Email system. The project had been in development for about two years since mid 2006. It has encountered a large amount of issues and it was suspended after a final review. The system implemented until then includes a web base user interface, triggers and special updating stored procedures in individual database. XML web service had been proposed during the development but had not yet been developed by the time the project was suspended.

4.4.2 Consideration for Research Questions

In some documents found by the research, the corporate directory project was thought to be something extended from the PerOrg project in the previous case, but it actually had a different set of objectives and been through a development life cycle with support from management. The project has its own user interfaces and interact with a number of sub-systems behind the scene. It is a complicated integration system of a number of separate and independent systems. The integration project was aimed for improving processes that are currently carried out separately on individual systems by different users from different departments. It would be of great benefits for the agency as a whole if such integration system could have been implemented successfully.

Although the project has been suspended at the end, it would still be an excellent case for the research on system integration and XML application. The project involved a few significant sub-systems, including Microsoft Active Directory, a commercial HR payroll system and an in-house intranet database. XML web services were identified and proposed in the system design document. The integration has a set of users' requirement from the beginning. The case will enable us to identify issues and solutions for such system integration in large organizations. The case study can provide answers directly related to those research questions.

Many large organizations would have systems like Active Directory for controlling the network, HR payroll system and Intranet. If they are successfully integrated, it will provide enormous benefits and process efficiency as they were stated in the business case. Although the project for this case is not a success, the research of it will help us to identify the causes for its suspension and solutions for future. If those challenges had been identified in the earlier stage of the project, they would have been better handled than they were and the project would have been a success. It will be very important and of great benefits for the agency that the case study can identify all those important issues and solutions.

XML web services were not mentioned in the beginning of the project, but they were found in some design documents created later on. The case study would aim to identify

how XML was involved in the integration project and to answer the research question. Initially the project was designed and implemented by a third party software company, but all source code and documentation have then been handed over to the agency. The corporate directory project might not be regarded as XML web service based solution, as XML web services were not included in the initial stage and were not developed when the project was suspended. However, the case study can still explore into why XML web services were included later and how they would have been used to help the project.

The corporate directory project has been in development for over two years and was finally suspended. The reason for such result has been the topic of investigation for the department and this research. There were a large number of documents, system design and source code produced during and after the project. These material including extra discussions with people involved will enable the case study to establish if XML web services were the critical success factor and why, which also answer the research question.

4.4.3 Summary of Case Findings

(Please refer to the appendix section for detail of findings and preliminary analysis)

- 1) The corporate directory project would be regarded as integration of common data and processes, with a few workflow elements. It was a complicate system beyond basic integration over 2-3 sub systems. It had involved a number of third party core systems including Active Directory, the payroll system and Microsoft Exchange server. All of the sub-systems are large and complicate. Although many workflow elements would be identified, but it had never been regarded as a workflow system, not even an integration system.
- 2) There were many factors leading to the suspension of the project at the end. The lack of clear business requirements from the beginning was identified as the primary factor. The project included integration of a number of sub-systems, which the development team was not familiar with or understood well. However, the project was regarded as a simple application to build within short period of time as indicated in the initial business case document.
- 3) The delay and the final suspension of the project were largely due to the project management issues as pointed out in the final report. From the detail analysis of

findings, there were many other important factors including the lack of proper system design.

- 4) The biggest issue in the system design was the interfaces among sub-systems. The initial design of using complicate stored procedures had caused major problems and delay. It has caused numerous issues for setting up for testing as pointed out in the final report. It would also be very difficult to maintain as they were hidden in the database. Although it was later changed to use XML web service, it was too late for saving the project. As the former system analyst said in the interview, the solution from the software company should have been evaluated properly before it was adopted.

Special characteristics of the corporate directory project:

- 1) The software company contracted to do the job did not to have some of the required expertise as pointed out by the final report.
- 2) The project was not identified as an integration project in any document. Common integration issues, such as data sharing, functional interactions were not found to be identified, documented and dealt with systematically during the entire life of the project, starting from the business case and system design.
- 3) There has not been a consistent design of the system. The original design was not been properly thought through initially, many changes were made during the project, including changing to XML web service interface, as amendment to the original design.

Current status for corporate directory:

- 1) The project has been suspended and most staff member involved with the project has left the agency.
- 2) New attempt in future is still possible, as no system has been found to be able to fill the gap that the project was aimed for.

Chapter 5 Case Study Findings and Analysis

5.1 Summary and Discussion of all Cases

5.1.1 What are the integration projects?

From business point of view

Case 1: It was to integrate a number of core business systems with the GIS system, so the business users can easily access and associate GIS information (maps) with other business data in a simple interface.

Case 2: It was to integrate the document management system with other core business systems, so the business users can easily create, access and associate document records with other business data in the simple interface.

Case 3: It was to establish data sharing of core organization data of office and person, so they are consistent and accurate across all business systems.

Case 4: It was to automate and simplify a set of processes in a number of existing systems by having a central user interface and a set of underlying process associated with those sub-systems.

Commentary

All cases had involved systems that can be often seen in large organizations, they were all aimed to integrate multiple systems and provide new features to the organization. As from the research findings, business systems were often related or connected to each other within the organization. Business processes that involve interacting with multiple sub-systems would be improved by implementing functional integrations using new user interfaces, such as in case 1, case 2 and case 4. Multiple business systems may share common data or require data from each other, such as in case 3.

From system development point of view

Case 1: It was to develop new interfaces, establish data sharing and enable functional interaction among a number of in-house systems with the specialist GIS system, which is a third party commercial software package.

Case 2: It was to develop and enhance the interfaces and components used by multiple in-house systems to work with Document Management (DM) system, which is a third party commercial software package.

Case 3: It was to set up central databases and establish sharing of office and person data. It also included using common software components to access those data for all in-house systems.

Case 4: It was to develop a new centralized user interface and a set of underlying processes across multiple sub-systems including MS Active Directory and HRIS systems, which are all third party software.

Commentary

All cases involved new developments to integrate multiple sub-systems in functional and data level. Of them, case 3 had the majority of the integration related to data sharing, and case 4 related to system process interaction. All cases had involved substantial amount of the development. As all in-house systems were built on the same backend database platform and they could either be integrated in database level or through other means, such as XML web service. In the agency, most in-house systems use the same operating system and software development platform, system integrations among them were much easier than those with third party software. In case 3, the integration was among in-house systems only. In case 1 and case 2, integrations were among in-house systems and third party software. In case 4, all sub-systems were from external vendors. Integrations with third party software were achieved in case 1, 2 and 4 by using dedicated interfaces, including XML web service and library component built in-house or supplied by the vendors.

5.1.2 What are the goals for the integrations?

From business point of view

Case 1: It was to enable business users to easily access and associate GIS information (maps) with their business data; it also provide users with new functions such as spatial query and others specific interactions between GIS and business systems.

Case 2: It was to fulfil specific business requirements including facilitating creation and association of document records with the business system.

Case 3: It was to ensure the core organization data of office and person to be consistent and accurate across all business systems.

Case 4: It was to improve the processes around setting up and maintain staff information in relevant systems, such as HRIS and Active Directory.

Commentary

Generally, system integrations can enable users to improve business processes and acquire data from multiple sub-systems more effectively and efficiently. Without adequate integration, individual systems can become islands of their own data and functions, which would hinder the functioning of the department as a whole. In summary, all projects had relevant business goals that could be regarded as high level goals as stated previously. High level goals need to be supported by analysis and functional requirements. In case 1 and case 2, functions requirements were specified clearly and documented in the business requirements document after extensive analysis of the system and the problem domain. In case 3, many brief attempt have been made on the project during a long period of time by various people, business goals were specified but without much involvement of users and hence were lack of support from the business and management. In case 4, only high level business goals were set out in the business case without proper analyst and business users' involvement before the project was started.

From system development point of view

Case 1: It was to use the latest technologies to integrate a number of in-house systems with a third party commercial software package GIS.

Case 2: It was to develop and enhance integration of multiple in-house systems with a third party commercial software package Document Management (DM) system by using the best available techniques.

Case 3: It was to enable effective sharing of the office and person data among all in-house business systems by developing new interfaces, central databases and common components.

Case 4: It was to integrate multiple sub-systems include MS Active Directory and HRIS systems, which are all third party software, by developing a new centralized user interface and a set of underlying processes.

Commentary

Generally, integrations requiring new developments would provide opportunities for adopting new technologies and further enhance existing systems. They can improve the organization IT architecture by making the sub-systems work together better or by reducing duplications on systems components and databases. In case 1 and case 2, the

latest technology at the time, ASP .Net and XML web services were adopted for the development of systems that were well specified and designed. In case 3, different technologies including SQL and .Net were used in different approaches. In case 4, the initial design consisted of old technologies carried over by the software provider, newer technologies such as ASP .Net and XML web services were only adopted after failure of those old ones, which was concluded in the mid-project review.

5.1.3 Why are they important to the organization?

Case 1: GIS was one of the essential systems for the department as the agency was responsible for manage most of public land of the country, so were the other systems involved. They were Permissions and Land Register. The integration has provided new powerful functions such as spatial query and enabled users to acquire GIS (maps) data directly from business system.

Case 2: DM was one of the essential systems for the department as it stores and manages all business documents, so were the other systems involved. The integration has enabled user direct access to DM and made it more effective in creating and maintaining business documents.

Case 3: Office and person are core organization data used by most in-house systems. The integration would ensure those important data are consistent and accurate across all business systems and the cost and effort to use them would be reduced.

Case 4: Systems like MS Active Directory and HRIS systems are basic and core systems for the agency, the integration would help to improve the relevant processes around them.

Commentary

All sub-systems involved were core systems for the organization as described in the individual case. All integrations would provide new and unique features to the organization, as they were argued in requirement documents or business cases. However, there were different among the cases, such as whether or not they were for core business systems, or for backend supporting systems. In case 1 and case 2, the main driver of the integration was from core business systems with specific requirements. In case 3, the integration driver was from overall IT structure for improving certain part of all in-house business systems, including core or non-core

business requirements. In case 4, the integration was for improving non-core business operational processes among some proprietary third party software.

5.1.4 What are the solutions that have been used so far?

	Integration System (Component)	Integration Platform (Technology)	Third Party Systems (Component)
Case 1	GIS Viewer DOCgis (customized product) Interface components (in Permissions and Land Register) Web service (for data and functional integration) Windows service (for regular data update) NGDB GIS databases	Microsoft ASP.Net Microsoft SQL XML web service	ArcIMS map service ESRI ArcSDE
Case 2	PermissionsDMEClient.dll DMESmartClient.EXE Interface components (in Permissions and Land Register) DM service DM Web Service	Microsoft ASP.Net Windows COM XML web service	DM Library (DME.EXE & CKO.DLL)
Case 3	PerOrg (in proposal only) Common (database) Office&Person Service Office&Person Component	Microsoft SQL XML web service	
Case 4	CorporateDirectory (database) Web Forms (User Interfaces)	Microsoft SQL Microsoft ASP.Net XML web service	MS Active Directory HRIS ARS MS Exchange (proposed only)

Commentary

All cases involved building new integration systems or components. By going through source code for all the cases, the researcher has found case 1 has the largest amount of development work (for building the new GIS Viewer systems, components and etc.).

Case 3 has the least amount of work; mainly due to the main PerOrg system had not been funded and developed. In case 4, most initial integration work was done within the database, then it was changed to use Web forms, some early work had become unused. In case 2, a small amount of early work had also been replaced by newer parts. All cases used some sort of Microsoft technologies, as the agency's IT platform is Microsoft based. All cases had used XML web service to various extents. In most cases (all except case 2), SQL server databases were used. Most cases (all except case 3) involved third party software and interface components. In case 1 and case 2, they were from ESRI and DM respectively. In case 4, there were multiple third party systems, including AD, HRIS and ARS.

5.1.5 What are the critical success factors for the integration projects?

Case 1:

- 1) Business requirements are agreed and well documented.
- 2) Suitable software and development tools are chosen.
- 3) Projects are well managed and supported.
- 4) Business Involvement
- 5) User Acceptance/Training

Case 2:

- 1) Clear Business requirements.
- 2) Suitable software and development tools and solution design.
- 3) Projects management.
- 4) User Acceptance/Training

Case 3:

- 1) Clear and basic business requirements
- 2) Management's support.
- 3) Suitable software platform.
- 4) Business Involvement

Case 4:

- 1) Clear business requirements.
- 2) Project Management.
- 3) Skilled resource and in-depth knowledge of the sub-systems.

4) Business Involvement

Commentary

The critical success factor (CSF) concept was first introduced by Rockart (1979) as ‘the limited number of areas in which results, if they are satisfactory, will ensure successful and competitive performance for the organization’. As argued in the Lam (2005) paper, CSF for Enterprise Application Integration (EAI) would be found in a number of areas, including Business (such as strong business case and requirements), Organization (such as top management support), Technology (such as using right EAI tools and technologies), and Project Management (such as realistic project plans and schedule). In summary, those most important CSF identified for all the cases in this research are very similar with only minor difference because of the different situation in particular case. They are largely in line with the result from Lam (2005) paper and the Standish Group (2004) report.

There was a long list for CSF as identified in literatures. For the integrations included in this research, the important CSFs identified were:

1) ***Business requirements are agreed and well documented.***

In both case 1 and case 2, there were clearly documented business requirements, which were result of in-depth analysis; especially in case 1, the process of requirement gathering had included prototyping and a workshop. In case 3 and case 4, the integration project had been in struggle because of the lack of clear requirements. In case 4, the lack of clear business requirements was identified as the primary factor for its suspension in the final report.

2) ***Suitable system design and development tools are chosen.***

In case 1, for the GIS viewer development, formal evaluation and proof of concept had been used to select software and decide on system design. In case 2, the development had evolved from traditional techniques (i.e. Windows COM) to more advanced ones (COM+, XML web service). In case 3, MS SQL was chosen because it was the common platform for all the in-house sub-systems. In case 4, the lack of suitable integration knowledge in the initial design was identified as a major factor for the project failure.

3) ***Projects are well managed and supported.***

For case 1 (GIS Viewer part), the project was well managed and the system

was delivered successfully. In case 2, the initial development was short of time and created some issues in the following system upgrade process, the project was then well managed and the integration system had evolved into being more effective and robust. In case 3, the initial project (PerOrg) was not supported, in the later approach (XML web service and .Net component), the project was only part of other existing system (Bioweb). If there were a proper project, a more satisfactory result would have been achieved. In case 4, proper project management was also missing, otherwise the project would not have been delayed for so long and problems would not have become irresolvable.

4) ***Good knowledge of the sub-systems***

This was identified by analysing the integration components and their results. In case 1 and case 2, there was third party software involved in both cases. The development team had put in a lot of effort getting to know and work with the third party software. The development team was also very familiar with the in-house systems to be integrated. The integrations were successfully developed. In case 3, although there was no third party software, there were a large number of in-house systems involved, the developer who had made the XML web service and .Net component was not familiar with other system except the one (Bioweb) he worked on. The solution had not been used by other systems. In case 4, there were a number of third party software from different vendor, the lack of in-depth knowledge in those systems would not had help creating a working system design, the situation was identified as a big factor for the project failure.

5) ***Business Involvement***

It was identified in all cases and was one of the main factors for the project success or failure. In case 1 and 2, because of involvement of the Permissions business user, particularly the business rep, the requirements were well defined. In case 3 and 4, the lack of proper business involvement was a major factor for the unclear requirements and not so successful outcome of the project.

6) ***User Acceptance/Training***

In case 1 and case 2, user acceptance and training were included formally in the project plans with supporting staff and material. It was identified as an important factor for the project success.

5.1.6 What are issues and challenges faced by the project team?

Case 1:

- 1) Business requirements were hard to defined and agreed on.
- 2) The database approach for the integration was not suitable in complex situation.
- 3) The off-shelf DOCgis product is difficult to maintain and lack of some crucial features.
- 4) A new approach needed to be chosen from many new GIS software packages and development tools.
- 5) The old GIS team was mostly isolated from the rest of development team.

Case 2:

- 1) The third party software was of old technologies.
- 2) Support was very limited from the DM vendor.
- 3) The initial development had only short time frame.
- 4) Software components have changed over time, including all sub-systems.
- 5) When there was a run-time problem in production, it was difficult tracked down.

Case 3:

- 1) The requirements of the office and person data vary for different systems and they would be difficult to define.
- 2) It was difficult to convince business to change existing systems without obvious benefits that can be seen from the user interface.
- 3) It was difficult to develop a simple solution to satisfy various business needs.

Case 4:

- 1) Business requirements were unclear and kept changing.
- 2) The system interfaces were not clearly defined.
- 3) The software provider has limited expertise development using latest MS technologies.

Commentary

As argued by Yimin Bao, Ellis Horowitz (1996), constructing new systems from collections of pre-existing third-party software and the commercial off-the-shelf products presented serious challenges. These challenges would come from many areas,

including management, business and technology, as they were illustrated in individual case above. The followings are those most commonly seen in this research:

1) ***Business requirements were hard to defined and agreed on.***

This was the primary challenge for most of the cases except case 2, which had its requirements specified clearly in a very early stage. It was found a lot of issues had been raised and also a lot of effort had been spent on defining business requirements in case 1, case 3 and case 4. In case 2, the business representative was very actively involved, she had written the users' guide and large part of the requirement. The requirements in case 2 were less complicate than those in the other cases.

2) ***There would be a lack of adequate resource for the integration projects.***

This was a challenge in many cases during different period. In most cases, it had direct impact on the projects and their solutions. In case 1, in the early DOCgis stage, the GIS team with only one developer had not capacity to develop new integration, as result, off-shelf products were chosen and 'quick and easy' database linkages were used. In case 2, the initial development time was too short for developing a proper solution. In case 3, there wasn't enough resource and management support (funding) mainly because of the lack for basic business requirements; the initial project (PerOrg) was never developed. In the later stages, XML web service and .Net component were developed within the scope and resource of a single business system (Bioweb). Case 4 was the only exception because the then CIO authored the initial business case with budget allocated.

3) ***There would be lack of knowledge for sub-systems.***

This was a challenge in all cases to various extents. In case 1, because GIS is specialist software, the development had a lot of difficulties in choose the right software tools and packages. There was evidence that the original GIS developer had not been involved in any of business systems. In case 2, the DM library was of old technologies. It had taken major effort to develop the initial integration and to further improve it in later stages. In case 3, The PerOrg proposal had not taken into account of different usage of the office and person data in different business systems, such as Permission and Bioweb. The design of the XML web service and .net component was mainly considered for the use of Bioweb only. In case 4, the development team had major issues with sub-

systems like HRIS and Active Directory because they had not known them well and had not anticipated those issues.

4) ***Different parties would manage sub-systems and they would be less willing to co-operate.***

In case 1, the GIS team had been very persistent with its existing backend database integration and DOCgis approach, their support to the new GIS Viewer had been limited and short-term. The GIS Viewer would have been developed faster and better if full support from the GIS team had been provided. In case 2, trying to get technical support from the software vendor had been proven difficult and time consuming. This was reflected during the investigation of issues with the DME.exe component. In case 3, during the analysis of PerOrg project and development of the XML web service and .Net component, there was very little or far less than enough involvement of people from other business system. In case 4, both HRIS and the Active Directory were managed by their own operators, their reluctant support to the project have contributed to many of the problems.

5) ***There can be a lack of governance and ownership.***

The issue was mainly of concern to the management of the agency. In case 1, the GIS team had strong ownership of the old DOCgis application; it had been kept in use even without some of the essential features. There was lack of ownership for the GIS Viewer after it was initially developed and deployed for the Permissions system. It would had been further enhanced and integrated with more business systems if it had a more active governance and ownership. In case 2, the slow progress of getting response from the vendor was partially because of the lack of governance and ownership of the integration system. The vendor would have responded much faster if the manager who was responsible for the purchase of the software could have been directly involved. In case 3, there was not much governance and ownership in the early stage for the PerOrg project as it was not funded, many people have worked on it with different approached as it was concluded from the various documents. For the development in the later stage, the governance and ownership of the XML web service and .Net component were only resided in an individual business system (Bioweb), which was not adequate for a integration relating to all other business system, as a result, the solution was not widely accepted. In case 4, it

was identified as the major factor for the project failure. The lack of governance was the main reason for causing many of the other issues, including the changing status of the requirements and other project management issues.

5.1.7 What are the major technical issues and solutions for the integration projects?

Case 1:

- 1) Business requirements were hard to defined and agreed on.
Business requirements had been clarified and defined through using prototype and workshop.
- 2) New development platform and tools needed to be chosen from many GIS software packages and technologies.
New tools were chosen through evaluation and proof of concept.
- 3) New approaches need to be identified for data and functional integration across multiple systems.
New XML web service and other XML configuration file were adopted for the integration solution.

Case 2:

- 1) The third party software was built in old technologies.
The development had spent a lot of effort to get familiar with all the components (library *.dll).
- 2) Software components have changed over time.
The interface component had been updated accordingly and XML web service had been adopted to interface systems from different version of .net framework.
- 3) When there was a run-time problem in production, it was difficult to track down.
XML web service had been adopted to better handle the server component.

Case 3:

- 1) Missing of clear and basic business requirements.
The issue had not been resolved because of the lack of analysis.
- 2) Searching for suitable software platform.
Various approaches had been attempted; the likely solution would be

combination of shared database and XML web service and all sub-systems having interfaced with them.

Case 4:

- 1) Business requirements were unclear and kept changing.
The business analysis and user had become more involved.
- 2) The system interfaces were not clearly defined.
New XML web service had been designed but not yet complete in the end.
- 3) The software provider has limited expertise in HRIS and Active Directory sub-system.
The agency took over the development by using a contracted developer.
- 4) The software provider has limited expertise development using latest MS technologies.
The agency took over the development by using a contracted developer.

Commentary

From the summary above and detail description in the individual case (in appendix), the followings were concluded:

- 1) ***Business requirements could be difficult to defined and agreed on.***
As it was argued in Lam (2005) and Standish Group (2004) report, clear business requirement was one of the top issues in IT development, especially for system integrations, because an integration system would likely interact with more other systems. There were some complex interactions with sub-systems in most of the cases in this research. The issue has become more technical as it had heavily depended on the integration technologies and what they could offer to the business. In case 1, there have been a lot of difficulties for the users to comprehend the functions that the GIS integration system could provide. Over a year had been spent for clarifying the requirements. In the end, it was successfully resolved by creating a working prototype and presenting it to the users, which had required a lot of the technical effort. A simply mock-up prototype would not work in that case, it would not attract interest from users and stimulate further thoughts for the requirements. Without a working prototype, any business requirement would be in danger of being impossible or too expensive to fulfil. This was reflected in case 4, it was mainly contributed by the initial designer's not knowing well the technologies of the sub-systems.

According to the initial business case, the requirements were thought to be simple and the project would be completed in three months. In fact it had been dragged on for over two years. Unsettled requirements were the key factors for the project failure as identified in the final report, but one of the underlying reasons was the lack of knowledge in the technical aspect of the sub-systems, although it was cited as a separate technical issue in the final report and other document. The technical factor for business requirement was also reflected in case 2 and 3. In case 2, the initial requirement for creating document entirely on the server without user interaction on the client was proven impossible to achieve. The requirement had then been adjusted accordingly. In case 3, because of the lack technical knowledge in the other business system like Permissions, the requirement for the XML web service and .Net component was unworkable for other systems like Permissions. In summary, a solution for the issues would require sufficient knowledge for the sub-systems and technical skills to help users to understanding the integration, such as by building a working prototype. Case 1 and 2 were examples with some success, and case 3 and 4 had provided lessons to learn.

2) ***It would be difficult to have known sub-systems well enough for the integration.***

This was evident in all cases. In case 1, this was a major issue in both project DOCgis and GIS Viewer. The original developer from the GIS team had not been involved or familiar with any of the business system. For the GIS Viewer, the developer was original from the Permissions system, it had taken him considerable effort to become familiar with GIS from a series of processes, including evaluation and proof of concept. In case 2, the original integration development had almost running out of time partially because the lead developer had not allocated enough time for the developer to get to know DM system. Although the solution was delivered in the end, but considerable improvement had to be added in later stages. In case 3, the analyst involved in PerOrg and the developer for the XML web service did not know other sub-systems well except the one they had work in, as there were so many other systems involved. In case 4, it was also identified in the final review and other records that the lack of knowledge of the HRIS and Active Directory from the

software provider was a major issue.

From the research, there wasn't any easy solution for resolving these issues except spending time and effort by the development team and recruiting more experienced staff. One of the reasons for causing these issues was that the systems to be integration were very different, both in term of technology and functionality. For example, in case 1, GIS software is specialist software; it is hardly accessible to ordinary users or developers. In case 2, DM was not a particular popular product; the developer would not have known anything about it before the integration. In case 3, although the sub-systems were of the same development platform for the same organization, but there were still major difference, for example, the Bioweb system only need office and person data for dropdown lists, but Permissions had use them to enforce security measures in most of its processes embedded in stored procedures. In case 4, the software provider had tried to overcome this with using another third party software ARS to interface with Active Directory, but the ARS component itself had become an issue.

3) ***It could be difficult to choose the right integration tool and technique.***

One of the most important factors in the integrations is choosing the right tools and technique to do the job. In case 1, without much of the choice at the time, DOCgis was chosen from the product list of the GIS software vendor. It did do its job and help to create the DOCgis solution, but it could only work for those simple requirements. It became inadequate when advanced integration requirements were raised. The development team had spent a lot of effort and evaluated a number of options before deciding on using the .net development platform to develop the new GIS Viewer in-house. The development team had done a number of proof-of-concept systems, which were the essential way for evaluating options. In case 2, for improving the integration system during the changes of the DM upgrade, various techniques including XML web service, COM+, and new CKO.dll library had been adopted. In case 3, there has been argument of whether or not using the common database approach, although it was pushed through by a lead developer at the time, after she left, the XML web service and .net component were choose and developed. In case 4, the default integration tool and techniques were the major factor for its delay; the

software provider bought them over from other systems. They simply not work for the department. They were largely replaced when the agency took back the bulk of the development work.

The causes for the issue varied. In case 1, there were a number of product can be used for integration in the market, but because GIS was specialist software, the developer would not have sufficient knowledge of all of them before hand. In case 2, DM was of old fashion client-server technology different from other business system. In case 3, there were too many system involved but not enough input from developer from other systems, even in the later stage of the component development. In case 4, it was found the software provider had no sufficient development knowledge.

In the research, the using of proof-of-concept was found an effective method to resolve the issue in case 1. The iteration release approach was also useful in case 2, as the integration had eventually become well-performing and also adopted for the changes of the DM system. There wasn't a solution in case 3, but open discussion and system demo would have helped getting other developer involved and better chance to make some informed decisions on choosing the development tools and techniques. In case 4, the initial mistake on the development tools and techniques was partially caused by the lack of new development knowledge as identified in the final report. The issue was only resolved by recruiting a new contract developer to do the job. In summary, the methods for resolving the issue would include

- i. To use an effective evaluation method such as proof of concept used in case 1.
- ii. To use an adaptable development method such as iteration release in case 2.
- iii. To allow more input from other developers using discussion and demos as in case 3.
- iv. To select developer with sufficient development knowledge, such as in case 4.

4) *An effective integration system would be difficult to design.*

Designing a solution for integration would be the biggest technical challenge, which was illustrated in all cases in this research. In case 1, the architecture of using DOCgis and relying on the backend database connection had been promoted by the GIS team for many years as the best solution. The chance of rethinking only came when it could not satisfy the integration requirements from Permissions system. It had taken many months of effort of an experienced .net developer with thorough knowledge of the business system to come up with the design of the new GIS Viewer. In case 2, the initial solution worked in a 'quick and dirty' fashion as described by its original developer, as there wasn't enough time for building a more robust one. The integration had been through much modification and enhancement since then. In case 3, the initial PerOrg system was simply not been formally designed or developed, as it required more funding than just some spare time from a few analyst. Although the XML web service and .net component worked for the Bioweb system, they could not be used for other systems like Permissions. In case 4, the initial solution just would not work after a long delay; the modified solution also could not be completed at the end.

The challenges for creating integration solutions would come from many directions, including those from management and human factors. In case 1, the specialist GIS software has created a major learning curve. To overcome the existing DOCgis shortcomings was also a major challenge for the GIS Viewer. In case 2, DM was totally different software from other business systems in the agency, and there was not enough time for initial development as claimed by the developer. In case 3, apart from the large number of the system involved, there wasn't any funding for the design and development, which was caused by management. In case 4, the initial solution was not workable because the software provider did not have sufficient knowledge as pointed out in the final report.

The research has found various approaches for creating integration solutions. In case 1, the GIS Viewer solution was in-house redevelopment of the old DOCgis in .net using XML web service and other XML technologies. It took a

lot of effort for the development team to create the solution by using a number of methods, including formal software evaluation, proof of concept, prototyping and user workshop. In case 2, the solution had been refined through a series of changes. Many versions the software had been developed as identified in the source code. XML web service was adoption in a few places to improve the design and overcome compatibility problem. In case 3, many small solutions were created after the PerOrg proposal, including the common database, the XML web service and .net component. In case 4, the solution was amended after the mid-project review. XML web service interface to the HRIS system was adopted, as shown in the agreement email message of for its development.

5.1.8 How has XML been used and to what degree?

Case 1:

1) GIS server interface

XML is the data format for communicating to the GIS server.

2) XML web service for data integration

The web service was responsible for providing GIS data in XML format to business systems (Permissions and Land Register).

3) XML web service for functional integration

The web service was responsible for enabling interaction between the GIS Viewer and business systems.

4) XML Configuration

A number of XML files were used in the GIS Viewer system for allowing dynamic data integration from business systems.

Case 2:

1) XML web service for data processing

The web service was responsible for processing data sent from the client side control and saving it to the database.

2) XML web service for providing platform independent interface

A web service was developed to enable communication between components with incompatible .net version.

Case 3:

1) OfficePerson XML Web service

It had been used by the Bioweb system for accessing the office and person data.

Case 4:

1) HRIS web service

The web service was the interface to the HRIS system.

Commentary

XML and web service had been used in all cases in various degrees as mentioned above. In case 1, XML was one of the core techniques for the GIS Viewer system. XML had been used in many essential parts of the integration system, including the GIS server interface, data integration, functional integration and the system configuration. The GIS Viewer system was essentially designed around the XML technology. In case 2, XML web services were only added after the initial solution. The XML components were used to improve the data processing of the integration and providing integration interface for systems with different platform requirement. The XML web service had made the integration more robust in the end. In case 3, XML web service was developed for data sharing. It was used by one business system only and also later wrapped up by a .net component. In case 4, XML web service was later chosen to replace the unworkable database backend access approach for the interface of the HRIS system. The use of XML web service was part of attempt to save the project but was not fully developed in the end.

5.1.9 What are the reasons for adopting XML in the projects?

Case 1 (XML was extensively used in GIS Viewer integration):

- 2) The GIS server interface (ArcIMS .Net Link library) uses XML data format.
- 3) The agency and the development team were more ready than before for using XML.
- 4) The XML based design was chosen through using formal development methodologies.
- 5) The GIS viewer development team had learnt from issues in the previous integration system without XML.

Case 2:

- 1) XML web service was a better option for data processing.

- 2) XML web service would be used by systems regardless their software platform or its version.

Case 3:

- 1) XML data would be used by other systems.

Case 4:

- 1) XML web services were adopted to replace the initial problematic design.

Commentary

According to the literature review, advantages of XML may include its open text format and platform independence nature as it was argued by Zachary G. Ives, A. Y. Halevy, D. S. Weld (2002); disadvantages may include extra overhead for processing XML data and being less efficient in data storage and query process than traditional relational data as it was argued by Margaret G. Kostoulas, Morris Matsa, Noah Mendelsohn, Eric Perkins, Abraham Heifets, Martha Mercaldi (2006). It is important these arguments could be applied in practice and desired results would be achieved. There would be many other reasons for why the integration solutions have chosen different techniques in different stages. Many factors must be considered for successful application in a practical environment. For example, there would be environmental and historical reasons, such as the lack of mature XML development tools and expertise at the time. The reasons identified in the research were:

- 1) When XML was already used by the sub-system, such as in case 1, the integration development had to use XML to communicate to the sub-system.
- 2) XML web service was regarded as a better interface options (including sharing by multiple systems) in all cases.
- 3) XML web service was adopted for resolving system incompatible issues, such as in case 2.

5.1.10 What are the reasons for not adopting XML in the projects?

Case 1 (XML was not used in the first DOCgis integration):

- 1) There were only a few dataset to be integrated (imported) to the GIS databases at the time.
- 2) There wasn't any resource for doing more analysis or development.

- 3) The DOCgis was customized from an off shelf map viewer product from ESRI and it was not designed for using XML for integration.

Case 2:

- 1) The document creation must use client side ActiveX control.
- 2) The saving of document linkage function was not designed properly.
- 3) The DM server could interpret document hyperlink URL.

Case 3:

- 1) The PerOrg project was designed as traditional client server application.
- 2) Common database for office and person was regarded as the simplest and easiest way for the integration.
- 3) DLL components were regarded as much easy to use.

Case 4:

- 1) The software company would not have the required skills.

Commentary

In summary, the reasons for not using XML or considering it in some cases or during some phases of a case were:

- 1) The off shelf integration system and sub-system were not designed for using XML, such as in case 1 and part of the case 2.
- 2) Direct database access and integration were regarded as the simplest and easiest way, as in case 1 (DOCgis) and common database in case 2.
- 3) There would be lack of skills and knowledge, such as in case 1 when DOCgis was developed and in case 4 when the initial designer came up with the initial design.

5.1.11 How were the decisions of the system design made?

Case 1:

- 1) Business requirements had forced the use of new design and techniques.
- 2) The agency's software development standard had affected the design.
- 3) The development team had followed formal development methods.

Case 2:

- 1) The original integration was done by a contracted developer under pressure to just get it delivered in the shortest time.

- 2) The development later had included improvement work on the original design.
- 3) Future consideration was also included in the latest work for the integration.

Case 3:

- 1) Business analyst made most of the decisions.
- 2) Developers would also push for their solutions.

Case 4:

- 1) The initial design would be from previous job by the software company.
- 2) New design was chosen because the initial design would not work.

Commentary

There was no consistent process identified in the research for making design decision:

- 1) In the early stage of case 1 and case 2, and all stages in case 3, technical decisions were made on Ad-hoc base by individual developer or analyst .
- 2) In the later stage of case 1 and case 2, developers had tried to follow guidelines such as the agency's development standard and use formal methods such as proof of concept and prototyping.
- 3) Design would be simply copied from previous work, such as in case 4.
- 4) Decision making would also be reactive. In both case 1 and case 4, new design had to be adopted when the old one had failed.

5.1.12 How successful were integration solutions with or without XML web service and other XML technologies?

Case 1 (with and without XML):

- 1) The old DOCgis without XML was not able to meet the new integration requirements; however, it is still in use, for which the research had found no technical reasons for.
- 2) The new GIS Viewer with XML had been well received by business and become one of the core systems with similar number of regular users as other major systems.
- 3) The XML based design was proven easy for integration when it integrated another business system with only minimum cost.

Case 2 (with and without XML):

- 1) The integration function had been core part of the business system and had been used to create a large number of records (near 20,000).

- 2) The old design without XML had been replaced by new one with XML web service for better design and robustness.
- 3) The integration with XML had expanded to include another business system successfully.

Case 3 (with and without XML):

- 1) The common database without XML approach has been used by a few systems, including Permissions, Intranet
- 2) The XML web service solution was replaced by the .Net component.

Case 4 (with and without XML):

- 1) The initial design without XML web service interface was not successful at all.
- 2) The new XML web service design had not yet been completed when the project was suspended.

Commentary

The research found mixed results from using XML and XML web service:

- 1) XML technologies had been used successfully in case 1 and case 2. In both case, the integrations had fulfilled the requirements and proven by statistic records.
- 2) The XML attempt in case 3 was not successful and was replaced by .Net DLL components.
- 3) The new XML web services design in case 4 was chosen to replace the failed initial design but not yet proven.

5.1.13 What are the critical success factors for the XML application(s)?

Case 1:

- 1) There were powerful and mature XML and web service development tools.
- 2) The development team has adequate (high-level) skills and experience, with support from the management.
- 3) The development followed methodologies.

Case 2:

- 1) Integration with third party software required more skills and effort.
- 2) System design should follow best practice.

Case 3:

- 1) There should be comprehensive design objectives.
- 2) Advanced XML Technique should be used when necessary.

Case 4:

- 1) Web service should have been included in the initial design.
Other than that, it was difficult to tell as the XML web service was only included in later stage of the project.

Commentary

In summary, the research found the following critical success factors for XML applications in the cases:

- 1) ***XML development tools***

In case 1, the new XML based design would only happen when the necessary tools were available. In case 3, the research had found not advanced XML development tools was used or available at the time.

- 2) ***Development Skills***

In DOCgis project of case 1 and the initial design of case 4, the lack of required development skills was identified as major factors for the system failure. In GIS Viewer and case 2, the high skill level of the development team would have led to the success of the projects.

- 3) ***Development Methodology***

In the GIS viewer of case 1, methodology was followed and good result was achieved. In case 2, the initial development was been pushed without proper design, the system had to be fixed afterward. In case 3, the ad-hoc design without thorough analysis led to the web service being unusable for other systems. In case 4, no initial analysis was done as it was concluded in the final report and the initial design was suspected to be just a carry-over from previous work.

5.1.14 What are the technical reasons for successful / not successful results for integration with/without XML?

Successful Cases

Case 1 (with XML in GIS Viewer):

- 1) XML and web service technologies have become mature and readily available for the GIS Viewer development.
- 2) The development team had managed to clarify the integration requirements.
- 3) The development followed methodologies.

Case 2 (with XML web service):

- 1) The design followed the best practice.
- 2) The XML web service is platform independent.

Case 3 (being somehow successful with common database without XML web service):

- 1) Direct access to the database was easy to set up.
- 2) Business systems like Permissions need database referential integrity.

Not So Successful Cases

Case 1 (without XML in DOCgis):

- 1) There was little development capability for the DOCgis at the time.
- 2) The DOCgis could not integrate data itself but rely only on the database interface.
- 3) The backend databases integration had created many issues for the database servers.

Case 2 (without XML web service):

- 1) It was a 'quick-and-dirty' approach.
- 2) Using client side ActiveX control to create Word document was not the best way it would be developed.

Case 3 (with XML web service):

- 1) The XML web service was slower than direct database access.
- 2) The XML interface functions were too simple.
- 3) XML data lacked database referential integrity feature.

Case 4 (without XML web service):

- 1) There was lack of understanding of the integration nature of project.
- 2) The initial design is just an assumption carried over by the software company.
- 3) The software company has not in-depth knowledge of the sub-systems.
- 4) The software company only has limited MS expertise required for the development.

Commentary

In summary, the technical reasons for the integration with XML being successful would be:

- 1) XML and web service technologies have become mature and readily available for the GIS Viewer development. (case 1)
- 2) The development followed methodologies and best practice. (case 1 and case 2)

The technical reasons for the integration without XML being somehow successful would be:

- 1) Direct access to the database was easy to set up. (case 1: DOCgis for previous integration; case 3 for Permissions)
- 2) Business systems like Permissions need database referential integrity. (case 3)

The technical reasons for the integration with XML being unsuccessful would be:

- 1) The XML web service was slower than direct database access. (case 3)
- 2) The XML interface functions were too simple. (case 3)
- 3) XML data lacked database referential integrity feature. (case 3)

The technical reasons for the integration without XML being unsuccessful would be:

- 1) The off shelf product (DOCgis) was designed to use direct database access. (case 1)
- 2) The backend databases integration had caused issues for the database servers. (case 1)
- 3) The development team would have no enough time to think it through but had to do it 'quickly'. (case 2)
- 4) The development team has no adequate expertise and had not designed the solution properly. (case4)

5.2 Analysis of the Findings

The following analysis was based on the above findings. It was aimed to establish answers for the research questions.

5.2.1 What were the integration solutions?

Top Category	Secondary Category	Supporting case evidence
Web	Database	In case 1, the old DOCgis was a web application with integrated data directly from database. In case 4, the initial design was also a web-based system with direct access to the backend database.
	XML Web service	In case 1, the new GIS Viewer was a web application using XML web service to integrated business data from different databases. In case 2, the DM integration (server part) use XML web service to communicate with DM server. In case 4, it was later changed to a web system with XML web service talking to the HRIS system.
	Third party component	In case 4, the web system also talked to the third party ARS component to interact with Active Directory.
Windows	Standard UI	In case 2, the document creation was done by a Windows UI component (DMESmartClient.exe).
	Background Process	In case 1, a Window service was created and deploy on the server to run regular GIS data update for Permissions and Land Register systems.
Database only	Tables and views	In case 1, the NGDB database was set up with views linking tables from other business databases. In case 3, the common database was set up to be used by other business databases.
	XML web servcie	In case 3, the office and person XML web service was developed for the data sharing.

5.2.2 Business requirements would be difficult to define.

Top Factors	Secondary Category	Supporting case evidence
Requirements were from various areas.	User Interface	<p>In case 1, new interfaces included DOCgis, GIS Viewer and various changes in Permissions and Land Register screens.</p> <p>In case 2, there were the new document creation screen and various screen changes in Permissions</p> <p>In case 3, a whole new set of screens were proposed for maintaining the office and person data in PerOrg project.</p> <p>In case 4, a whole new set of web forms were proposed and built for maintain the staff info.</p>
	Data	<p>In case 1, data from various business systems was combined with GIS data and display in the reports. GIS data were also used to update business data regularly.</p> <p>In case 2, business data was passed to the integration for creating new documents.</p> <p>In case 3, office and person data was shared among business systems.</p> <p>In case 4, staff personal and network details were stored in various systems, but maintained by the integration.</p>
	Functions	<p>In case 1, new functions in GIS Viewer included users defined spatial query on the map and dynamic linkages between GIS and the business systems.</p> <p>In case 2, they were document creation and saving to the DM server. The function of dynamic checking of document details was included later.</p> <p>In case 3, the PerOrg system would have a whole set of function for maintaining the office and person data.</p> <p>In case 4, the web forms had a whole set of functions to maintain staff details.</p>
	Process	<p>In case 1, new process was developed to keep the data in GIS database and business systems in sync.</p> <p>In case 2, integration replaced the manual process for creating documents.</p> <p>In case 3, the ProOrg would have changed the process for setting up users for the business systems.</p> <p>In case 4, all existing processes for staff management would have been encapsulated by the integration.</p>

They varied in scope.	Large	In case 4, the project covered many systems and complicate processes. It had the biggest budget found in the research and took the longest time of over two years of continuous development. In case 1, although it also covered many systems, large part of work was in database linkage for the old DOCgis. Most development work was done for the new GIS viewer.
	Medium	In case 3, the PerOrg system would be a medium size system to build comparing with others in this research.
	Small	In case 2, the initial integration was mainly to activate the DM function to create documents on the fly.
They would change during the project.	Major	In case 1, the change of requirement had force the GIS team to start the new integration development of GIS Viewer. In case 3, the project was changed from a full feature stand alone PerOrg system to simple data sharing.
	Medium	In case 4, the initial requirements had been substantially reduced in an attempt to make a delivery and the requirement for the HRIS system were consolidated by the business owner.
	Small	In case 2, the integration had to cope with version changes of the DM software and addition of new business system. New dynamic document checking function was added later.

5.2.3 In-depth knowledge of sub-systems would be difficult to acquire.

Top Factors	Secondary Category	Supporting case evidence
Systems would be from external suppliers.	Specialist Software	In case 1, GIS server include the ArcIMS and SDE engine were supplied by the GIS specialist software provider. Development used to be done by the dedicated GIS developer and external contractor. For the GIS Viewer, the internal developer had spent major effort to understand the GIS server and to develop the integration system.
	Traditional but outdated	In case 2, the DM software was built in traditional client-server design using Windows COM component library.
	Proprietary	In case 4, the HRIS and Active Directory were proprietary systems. They had their unique design and technologies. They were designed for particular business requirements (HRIS) or for the underlying network structure (Active Directory). Integrations features were generally not included in these systems.
	Too many of them involved	In case 4, the initial business case included many other systems which were excluded in later stages. They were MS Exchange server and the telephone directory system.
Systems were developed internally in the past.	Different technology	In case 2, Permissions and Land register system were of different .Net version. In case 3, systems included in the PerOrg proposal were built in different technologies. For example, Bioweb was of ASP, VB6 and COM component. Permissions was of .Net platform.
	Different Software Design	In case 3, systems would use the office person data in very different ways. Bioweb used them for dropdown lists only, but Permissions embedded them in its security model.
	Too many of them involved	In case 3, the original PerOrg proposal included over a dozen systems in its architecture diagram. They were of very different technology and system design.

5.2.4 Integration technique would be difficult to define.

Top Factors	Secondary Category	Supporting case evidence
<p>They would need to be learnt before and during the development.</p>	<p>They would require specialist software</p> <p>Systems used traditional but outdated technologies.</p> <p>Proprietary systems would be self protective and difficult to integrate with.</p> <p>Sub-system used various technologies and designs.</p>	<p>In case 1, the GIS Viewer relied on the ArcIMS to provide all the GIS data (maps). Special functions for handling maps and other GIS data were built into the integration system, particularly for spatial queries.</p> <p>In case 2, the DM integration had to communicate with the old fashion COM library from .Net platform, and even embedded ActiveX controls in the HTML page.</p> <p>In case 4, the integration had to relay on another third party component ARS to communicate with Active Directory. The initial design of using backend access to the HRIS system failed and was replaced by XML web service interface later.</p> <p>In case 3, the integration had to included direct database access for some system design requirement, such as Permissions. The XML web service approach would only be adopted by system like Bioweb.</p> <p>Also in case 2, XML web service was built for accommodated systems with different .Net version.</p>

Development tools would be unavailable.	They depend on the development platform	In case 1, the DOCgis system was a customization of an off-shelf product. Not design changes could be done to it and it was found not suitable for the new integration requirements. The new GIS Viewer was chosen for the .Net platform as it was used by the agency and the expertise had been accumulated and available. In case 2, 3 and 4, XML web services were included later in the integrations partly because they were easy to develop in the .Net development platform.
	Technique was relatively new.	In case 3, the XML web service integration contained only simple functions because advanced XML development tools such as Xquery were not available at the time.

5.2.5 Issues would come from the organization and project management.

Top Factors	Secondary Category	Supporting case evidence
The agency management might not support the project.	<p>The requirements were not clear.</p> <p>There would be no clear business owners.</p> <p>The management would have other goals.</p>	<p>In case 3, the requirements of the PerOrg system and the office and person data sharing requirements were never completed or properly done by analyst.</p> <p>In case 3, there was also no clear business owner for the office and person data and the PerOrg project.</p> <p>In case 1, the former GIS manager was unsupportive for new the GIS Viewer to replace the DOCgis, although the DOCgis was old and would not be improved much.</p>
The project might not be managed properly.	<p>There was a lack of governance of the project.</p> <p>The project would have been driven mainly by individual instead of proper plan.</p>	<p>In case 4, the project was allowed to delay for a long time, a 3-month project had ended up lasting for over 2 years.</p> <p>In case 2, the original developer was under pressure to deliver a ‘working’ version of the integration and to leave things to be fixed later.</p> <p>In case 3, the common database was driven by the lead developers of Permissions system, and XML web service by Bioweb developer.</p>

5.2.6 Business involvement would be difficult to get.

Top Factors	Supporting case evidence
It would take a lot of time and effort.	In case 1, it took over a year for the initial consultation with business for the GIS viewer project. The prototype workshop organized with a lot of dedicated effort including creating the prototype and organizing the venue.
Too many systems would be involved.	In case 3 and 4, there were a large number of systems involved for the integrations. Little documents were found showing effort had been spend on getting all the business people to be involved.
It would require business knowledge.	In case 4, the business representative involved for the first 6 months of the project was regarded having no clue of the business requirements.

5.2.7 User Acceptance and training would be missing.

Top Factors	Supporting case evidence
It was the last thing to happen.	In case 4, no consideration of user acceptance and training was identified in project documents.
It was not regarded as necessary	In case 3, the integration was mainly for developers and API, no end users would be involved. In case 4, the integration was regarded as a simple system only used by a selected group of users.

5.2.8 What were the methods to resolve the issues?

To clarify business requirements

Top Factors	Secondary Category	Supporting case evidence
To involve users and business owners	Business users would define the requirements.	In case 2, the requirements of the DM integration were clearly described by the user representative who also authored the users' guide.
	Business users would help to clarify requirements.	In case 4, the business owner was very involved in the later stage to reduce and clarify the requirements.
To apply proper system analysis.	Initial analysis should be done before development started.	In case 4, it was pointed out in the project review the development started badly by having no initial analysis at all.
	To allocate proper analyst resource to the project.	In case 1, a full time GIS analyst was allocated to the project and it took her over a year to develop a basic set of requirements. In case 3, no analyst had been assigned to the project for as a full time job, many documents were written by various analysts when they had spare time. No clear requirements were defined in the end. In case 4, a full time analyst was assigned after the project review in an attempt to rescue the development
To follow development methodology.	Developer would also help with the analysis.	In case 3, the lead developers of Permissions system, and the Bioweb developer would had clear idea what the requirements would be for their own systems for the integration.
	Using working prototype would be effective.	In case 1, the working prototype had helped to confirm the initial requirements written by the GIS analyst.
	Using workshop and user feedback would be effective.	In case 1, the workshop had generated lots user feedback and finally clarified the requirements.
	To allow iteration release and incremental development.	In case 2, the DM integration had been modified and enhanced over a long period of time.

		In case 3, both the common database and XML web service had provided solution for some business systems.
--	--	--

To acquire knowledge of the sub-systems

Top Factors	Secondary Category	Supporting case evidence
To use internal resource	To use developers who have more experience in sub systems	In case 1, the developer for the GIS Viewer had also done a lot of work for Permissions as found in the source code. In case 2, the second developer had also done a lot of work for Land Register.
	To discuss around internal developers	In case 3, developers had discussion about the issue and proposal in meetings. The developer for Bioweb had also demonstrated the XML web service.
	To allocated learning time in the project plan	In case 2, developers were allowed time to get familiar with the new version of the DM library in the upgrade and modification stages.
To use external resource	To use reference material provided by vendor.	In case 1, the documentation of ArcIMS from the vendor was used as reference. In case 2, the development reference for the DM vendor was used.
	To use third party component.	In case 4, a third party software ARS was used to reduce the complexity interacting with Active Directory.
	To direct communicate with the vendor	In case 2, teleconference was used to get technical advice from the vendor. In case 3, the HRIS vendor was consulted in many occasions.

To select integration technique

Top Factors	Secondary Category	Supporting case evidence
To use internal resource	To do software evaluation.	In case 1, evaluations had been done on a number of software products with consultation from providers.
	To do proof of concept on new techniques	In case 1, proof of concept had been done with a number of options. In case 2, proof of concept had also been tried on creating the document on the server side.
	To use experienced staff	In case 1, the GIS Viewer developer had done more work in .Net and other business systems than the original GIS developer for the DOCgis system.
	To follow methodologies and best practice.	In case 1, the development of GIS Viewer had followed through standard methods with document trail. In case 2, the modification and upgrade were done with best practice as explained in the findings.
To use external resource	To outsource the whole development.	In case 4, the development was outsourced to the software company, although they did not have the expertise as pointed out by the report.
	To outsource some component development	In case 4, the agency had agreement with the HRIS vendor for the XML web service to be developed by the vendor.
	To hire experienced contractor.	In case 4, two contractors had been hire for developing the web form interface.

To resolve management issues

Top Category	Secondary Category	Supporting case evidence
To get better management support	<p>To get project funded properly.</p> <p>To get business users involved.</p>	<p>Integration work in case 1, case 2 and case 4 were all fully funded, development had allocated resource.</p> <p>In case 3, the PerOrg project was not funded and project was abandoned. .</p> <p>In case 1, the business representative had written a large part of the requirements, because of her involvement, the business was the main driving force for the new GIS Viewer development.</p> <p>In case 3, there wasn't much business driver for the project except the lead Permissions developer at the time, most business systems were reluctant to change.</p> <p>In the end, there was not enough support for developing a complete solution.</p> <p>In case 4, there wasn't much business involvement initially as found by the review, which had hindered the development.</p>
To manage the project properly.	<p>To assign project manager with responsibility and accountability.</p> <p>To follow project management methods.</p>	<p>In case 1 and case 2, the projects were well managed by the project managers with many project documents found in the research.</p> <p>In case 4, no proper manager was assigned initially, many discussion records were found between various party discussing project issues with no solution. The project was delayed for a long time.</p> <p>In case 1, many project documents were found showing the project had regular meetings, issues were registered, discussed and resolved in meetings.</p>

To get business involved

Category	Supporting case evidence
To involve business from early stage	In case 1 and 2, business people were involved very early, in fact, they had requested for the integrations in the first place.
To provide prototype and run workshop	In case 1, the prototype workshop was held to allow more user involvement in the system analysis and design.

To ensure user acceptance/training

Category	Supporting case evidence
To plan ahead	In case 1 and 2, user acceptance and training were included in the project plans.
To ensure the requirements are clearly defined and agreed.	In case 1 and 2, the requirements were well defined as found in the documents. They were the base for creating the users guide and other training documents.
To involve business	In case 1 and 2, business users were heavily involved, the business rep had written the users' guide covering the new integration functions.

5.2.9 Why were they the critical success factors for the integrations?

Top Factors	Secondary Factors	Supporting case evidence
Clare and Basic Business Requirements	Management support	In case 1 and case 2, there were clearly documented requirements and the integrations were successful. In case 4, the final report has pointed out there was no initial analysis and requirement had kept changing, the project was suspended and integration not developed. In case 3, when the XML web service was developed, the developer only recognized requirements from his own system, the solution would not be used by other systems.
	Integration technique	In case 3, without clear requirement, the management would not support and fund the PerOrg project. In case 1, without the new integration requirement being raised by Permissions, the new integration technique and the GIS Viewer would not be started.
In-depth Knowledge of Sub-systems		In case 1, for the development of GIS Viewer, the developer had done evaluation, consultation and proof of concept with GIS software beforehand. The developer had also done a large amount of work for many business systems including Permissions, which was to be integrated with GIS in the GIS Viewer. The integration was successful. In case 2, the initial developer had put comments in the source code suggesting future changes that he had not got time to do. For the version upgrade afterward, the new developer had been given extra time to plan and to get familiar with the DM server. Although the initial integration worked, it only became better and more robust after dedicate effort for getting to know the DM was spent in the later stages.

	<p>Business requirement</p> <p>Integration technique</p>	<p>In case 3, the developer for the XML web service had not worked in and known the Permissions system, causing the service being unusable for Permissions. In case 4, the review report had pointed out in the final report, the software company had not adequate expertise and knowledge of the agency’s HRIS system.</p> <p>In case 4, without in-depth knowledge to the agency’s HRIS system, the initial business requirements were neither clear nor feasible. In case 3, without knowing the security design of the Permissions system, the requirement of the XML web service did not cover the need for the Permissions system.</p> <p>In case 4, without in-depth knowledge to the agency’s HRIS system, the initial integration technique relying only on database programming would not be feasible. In case 1, the integration technique was selected only after the developer had worked with GIS database for a few months and known the systems.</p>
<p>Adequate Integration Technique</p>	<p>Business requirement</p>	<p>In case 1, the GIS Viewer had used a new integration technique different from the old DOCgis, and successfully fulfilled the new GIS integration requirements. In case 2, although the initial integration worked, the integration component for document creation on the client side had costed major re-development effort for the change of the DM software. If it were designed better initially with future changes in mind as the final modification was, the work will be substantially less. In case 3, none of the common database, XML web service and the .Net component approaches worked for all the business systems. The missing of suitable integration technique was the core factor for the current state of the largely un-integrated systems. In case 4, the mistake in the initial design of the system was found to be the major factor for the delay and failure of the project.</p> <p>In case 1, without the adequate technique adopted, the development would not have been able to build the working prototype, to get feedback from users and to confirm the business requirements.</p>

<p>Management support and project management</p>	<p>Business requirement</p> <p>Knowledge of sub systems</p> <p>User/Business Involvement</p> <p>User Acceptance// Training</p>	<p>Management support and project management would have affected the integration projects in all cases, but to different degrees. In case 4, project management was identified in the final report as the No.1 factor causing the failure. In case 3, the lack of management support led to the project unfunded and un-developed. In case 2, lack of planned time for the integration work led to ‘quick-and-dirty’ design in the initial delivery. In case 1, the proper support and management had led to the success of the GIS Viewer project.</p> <p>In case 4, the lack of project management led to the initial development without clear business requirements.</p> <p>In case 1 and later stage of case 2, the developers were allowed extra time to acquired knowledge for the sub-systems by the project management as shown in the plan.</p> <p>In case 4, if the management would have ensured the business involvement from the beginning of the project rather than after the mid project review, the project would have been a success.</p> <p>In case 1 and 2, the project management had factored the time and budget for the user acceptance and training to ensure the success of the projects.</p>
--	--	--

<p>User/Business Involvement</p>	<p>Business requirement</p>	<p>In case 1, during the analysis of the new GIS integration project, extensive consulting with the business had been done by the GIS analysis over a long period of time. The prototype workshop was set up to maximized business involvement and feedback on the project. All these effort had led to the clear and basic requirements of the GIS Viewer.</p> <p>In case 4, the lack of business involvement from the initial stage of the project was identified and pointed out in the final report, it was one of the main factors leading to a failure of the project.</p> <p>Still in case 4, the lack of business involvement also caused the system requirement being fluid and unsettled during the development, which was also pointed out in the report and identified in other documents as one of the major faults in the project.</p>
<p>User Acceptance// Training</p>		<p>In case 1 and 2, user acceptance and training was included in the project plans. The project team had created special training documents for the new functions and dedicated trainers were employed to run training sessions for all users. The systems were well accepted and good usage statics was identified.</p>

5.2.10 What were the XML applications and the reasons for them in the cases?

Top Category	Secondary Category	Supporting case evidence
Data integration	Loose coupling	In case 1, XML web service was used to integration GIS data and business data. In case 3, the office & person web service was for data sharing among business systems. In case 4, XML web services were adopted later to interface with the HRIS system. In all cases, the XML web services were configurable and they were aimed to avoid the tight coupling of integration, such as object level linkages in databases. They would also be shared by multiple systems.
	Data process	In case 2, a web service was adopted to process data sent from the document creation component from the client, as it increased the flexibility and robustness of data processing and integration.
Functional integration	Third party software factor	In case 1, the GIS component ArcIMS already use XML for it request and response data, the GIS Viewer had to create and interpret those XML documents.
	Multiple platform	In case 2, a web service for the integration had to be created to communicate with a new system that was with a different .net version.
System settings	Web based systems	All web based integration systems included in all cases were ASP .net web applications using XML configuration in their web.config files. The files would allow direct and easy changes to the system settings.
	Integration Configuration	In case 1, the data for layer tree and spatial query set up for GIS Viewer was stored in XML files. They would allow easy change to these areas and were successfully used when Land Register was integrated with GIS Viewer.

5.2.11 What were technical CSFs for XML application in the integrations.

Top Factors	Secondary Category	Supporting case evidence
XML development tools		<p>In case 1, the success of the GIS Viewer would not be without XML and web service development tools being mature and readily available at the time, as opposite to situations when DOCgis was developed in 2002.</p> <p>In case 3, the agency had not advanced XML tools at the time, which would allow the developer to develop more advanced functions than just a few standard list data. The XML web service would have been more useful for other business systems otherwise.</p>
Development Skills		<p>In case 1 GIS Viewer and case 2, the XML and web service function were done by experienced developers, which would be an important factor for their success.</p> <p>In case 4, the lack of .net development skill of the software company was identified in the final report as a major factor for the failure of project and not being able to use XML web service.</p>
Development Methodology		<p>In case 1 GIS Viewer, the development had followed methodologies using techniques such as prototyping and proof of concept as found in the document trails, which would be an important factor for its success.</p> <p>In case 3, the ad-hoc design without thorough analysis led to the web service being unusable for other systems.</p>

5.3 Findings for the Research Questions

5.3.1 What would be the system integrations within a large government agency?

The research has found the following characteristics of system integration in the agency:

- 1) There were many kinds of system integrations within the agency. Many of the systems within the agency relate to each other in one way or the other. They would share common data, such as office and person data sharing case and the initial DOCgis integration. They would have closely related business processes, such as in DM integration and the corporate directory.
- 2) The research found integration requirements would be from four directions. They are user interface, data, function and process. All cases required new user interfaces as they were identified in the finding. All cases have data integration requirements, in case 1, GIS data (maps) was combined with business data. In case 2, business data was used to create new documents. In case 3, office and person data were to be shared by all business systems. In case 4, new and modified personnel data were to be passed to various systems. Two of the cases had obvious functional integration. In case 1, the spatial query function from the GIS server was combined with business query function in the new integration. In the case 2, the document creation function was combined from the Permissions and DM functions. Process integration was found in case 4, where it was aimed to automate relevant processes in the HRIS and Active Directory system.
- 3) The requirements of integration would grow over time, it would be important to design the solution to accommodate future changes. In case 1, the initial DOCgis system was not designed for system integration requiring more than just data, so the new GIS Viewer had to be developed to fulfil the integration requirements for Permissions. The GIS Viewer was designed using XML and for generic integration requirement. It was then able to be adapted to be integrated with another business system with only small effort. In the other case, the initial DM integration was built to meet the deadline without enough time to think through the design; it would have involved much less modification work if it was done right in the first time.

- 4) System integration projects are important for organizations as they would not only combine the features of the sub-systems, but also provide new features and satisfy new requirements. In case 1, the new spatial query features in GIS Viewer was a powerful tool for allowing users to define query on the map, it only worked with the integration of GIS and business system. In case 2, the DM integration automated the business document creation using business data. The corporate directory was also aimed for some important process automations.
- 5) System integration projects would be very complex and last for many years, as they would involve many different sub-systems of different internal designed, and sub-systems could change over time. For example, in case 2, the DM system had three versions changed. In case 3, the office and person data sharing, it involved over ten systems as illustrated in one of the architecture diagram, some of them like Bioweb and Permissions system had very different use of the same office and person data.
- 6) Not all of integrations were very big or would be regarded as enterprise application integration (EAI) comparable to those ERP package. Many of the integration systems would be regarded as small expansion of existing system; they would involve a small number of new interfaces. In case 2, the DM integration was a relatively small one with only a few interfaces.
- 7) System integrations would more likely to happen for systems with some sort of a common platform. In case 1, all systems used the same MS SQL database server. In case 3, all systems were in-house built with the same Microsoft development platform. In case 4, when the corporate directory tried to integrate with HRIS and Active Directory systems which were from different platform, it had encountered numerous difficulties. Although it was thought to be a three-month project initially, it dragged on for over two years. In summary, common platform would be an important pre-condition for the integrations found in this research, without it, the integration would be difficult.

5.3.2 What were the integration solutions?

Because the agency development platform was based on Microsoft technology, the solutions in the cases were all based on Microsoft technologies. The research found

integration solutions would be put into three categories: Web-based UI, Windows-based and database only.

- 1) Most of the solutions were of web-based systems with browser UI, as they were found in case 1, 2 and 4. In case 1 and 2, the sub-systems were also web-based applications. Within this category, the solutions would be further categorized into direct database access, XML web service and third party integration component. In part of case 1 and case 4, the web-based systems made direct access to the backend databases. In part of case 1, 2, and 4, the integration systems talked to the purpose built XML web services for accessing the backend databases or sub-systems. In case 4, the web-base system talked to the third part integration component for accessing the sub-system.
- 2) Integration solutions also included some Windows applications, which would be categorized into application with UI and background process. In case 2, the integration was achieved by creating an EXE running on client PCs with interaction with users. In case 1, a Windows service application was created and deployed on the server. It ran regular processes to make sure the data in the sub-systems were in sync.
- 3) There were integration solutions focusing on databases only. In part of case 1 and case 3, the solutions were made up of linking tables from different databases and creating special views pointing to common databases. In case 3, a solution was attempted by creating a special XML web service providing a shared interface for all the sub-systems.

5.3.3 What would be the critical success factors of these integrations?

There was a long list of CSFs as they were identified in the literature review. As the result of the analysis, the researcher would argue for the followings as the most important set of critical success factors found in this research:

Clare and Basic Business Requirements

Clear and basic business requirement for system integration would be difficult to define. The reasons were identified in the previous section. They are summarized below.

- 1) They would come from a number of directions, including user interface, data, function and process.
- 2) They would also vary in scope. Some would be relatively simple and basic, like case 2; some would be complicate, like case 3 and 4.
- 3) They would change during the project period, which were found in all cases.

From the case study:

- 1) When they were basic and clear, the integrations had succeeded, even with difficulties in development, such as in case 2 DM integration. In case 1, a lot of time and effort had spent on gathering the integration requirements, they were paid off when the new GIS Viewer was developed and well received by the users. Otherwise the projects would likely fail.
- 2) The business requirements would also affect other factors, including project management support and finding integration technique.

In-depth Knowledge of Sub-systems

Knowledge of sub-system would be difficult to acquire. The reasons include:

- 1) The sub-system required specialist knowledge and software, such as in case 1.
- 2) The sub-system was third party software of old design and technologies (client-service and COM interface), such as in case 2.
- 3) There were too many sub-systems and different design, such as in case 3.
- 4) The sub-systems were proprietary software with little access, such as case 4.

From the case study:

- 1) When the team had in-depth knowledge, the integration would succeed, such as in case 1 and 2. Otherwise the projects would likely fail.
- 2) The knowledge of sub-system would also affect other factors, including defining the business requirements and finding integration technique.

Adequate Integration Technique

Workable and robust integration technique would be difficult to identify. The reasons include:

- 1) They would need to be learnt before or during the development, the reasons included:

- i. They were specialist software, such as case 1.
 - ii. Systems used traditional but outdated technologies, such as case 2.
 - iii. Proprietary systems would be self protective and difficult to integrate with, such as case 4.
 - iv. Sub-system used various technologies and designs, such as case 2, and 3.
- 2) Development tools would be unavailable, as
- i. There were dependencies on the development platform, which happen in all cases.
 - ii. The technique would be relatively new at the time, such as case 3.

From the case study:

- 1) When they were identified and used correctly, the integration would succeed, such as in case 1 and 2. Otherwise the projects would likely fail.
- 2) The integration technique would also affect other factors, such as clarifying the business requirements.

Management Support & Project Management

Management support & project management would be the most crucial factor for many integration projects as they would be for other IT projects. In this research, they were identified as the major factor for failure of the case 4 corporate directory project in its final report. Corporate directory was the project with the biggest budget and longest continuously development effort in the case selection. In another cases, the office and person data sharing (PerOrg) project, they were crucial factors too. It was found that the lack of ownership of the office and person data, and the lack of support from the management were the major factor for the project being unfunded and un-developed. They were less of problem for other projects with clear requirements, such as case 1 and case 2. In the DM integration, even there was a project management issue of no enough initial development time being allocated, a working iteration was delivered in the end. In summary, the research found management support & project management would become more important when the integration project was big and requirement was unclear, as happened in case 4. On the other hand, they would also affect other factors, such as clarifying the business requirements and acquiring knowledge of the sub-systems.

Business Involvement

Since integrations would involve multiple systems, it would be difficult to involve the business people from all the system, but the lack of such involvement would make the integration requirements unclear and the system difficult to develop.

From the case study:

- 1) In case 1 and 2, the business was well involved and the projects were well developed. The integration involved a small number of systems including only Permissions and later Land register.
- 2) In case 3 and 4, there were a larger number of sub-systems involved, and business people for those system had participated a lot less than they should had. The requirements of the projects were not clearly agreed and documented, which was the main reason for the project failure.

User Acceptance/Training

Integration would provide new functions which would be complex, like case 1 GIS integration. In order to achieve the goals of the integration and the agency to reap the benefit of the new functions, training must be provided to users. User training would be expensive and have to be budgeted for, which would lead to the missing of it in the project plan. It also would only be done when the new system becomes available.

From the case study:

- 1) In case 1 and 2, training was provided by dedicated resource which leads to the good user acceptance of system. The systems had accumulated a large number of data created by the DM function and the sever statistics showed large number of user for the GIS system.
- 2) In case 3, the web service and .net component were used in a small number of systems by developers, it had little impact on the end users.
- 3) It was not applicable in case 4 either, as the systems were not completed.

5.3.4 What would be the methods for resolving these issues?

To clarify business requirements

Methods identified in the case selections were:

- 1) To involve users and business owners
As business people were the end users of the integration, they also have experience

and knowledge of using the sub-systems, in many cases in the research, they had and would had help the clarification of the business requirements. In some case they had defined some of the requirements almost by them self.

2) To apply proper system analysis

Analysis was the key factor for getting business requirements. From the research, it was found initial analysis must be applied before any development would start; projects must be have proper analyst assigned to; at last, experienced developer would also help with clarifying requirements.

3) To follow development methodology

There would be many methodology for system analysis. The research had found prototype was used effectively in one of the case; the development had also used workshop and user feedback; at last, by doing iteration releases, business requirements would be defined and fulfilled stage by stage in some cases.

To acquire knowledge of the sub-systems

Methods identified in the case selections were:

1) To use internal resource

i. To use developers who have more experience in sub systems

Some of the integrations were developed by developers who also work extensively with other business systems.

ii. To discuss issues and share knowledge around internal developers

Many documents and records have shown developers in the agency used demos and discussions to try to share system knowledge.

iii. To allocated learning time in the project plan

In some case, extra time had been allocated for the development team to get familiar with the sub-systems.

2) To use external resource

i. To use reference material provided by vendor

In many cases, documents were provided by the vendors and were the major source of knowledge for the sub-systems.

ii. To use third party component

In some cases, a third party components were evaluated or actually used for the integration to function as an easier interface to the sub-system.

- iii. To direct communicate with the vendor

In a few cases, the development team had to talk to the vendor directly for technical advice.

To select integration technique

Methods identified in the case selections were:

- 1) To use internal resource

- i. To do software evaluation

In case 1, a number of software had been evaluated before the final design decision was made.

- ii. To do proof of concept on new techniques

In case 1, proof-of-concept systems had been done for a few options.

- iii. To use experienced staff

In case 1, a more experienced developer was assigned to the new GIS Viewer development.

- iv. To follow methodologies and best practice

In some cases, development methods were followed and documented for optimal result.

- 2) To use external resource

- i. To outsource the whole development

In case 4, the whole development was originally outsourced and expected finished in a short time.

- ii. To outsource some component development

In case 4, some of the development was still done by the software company and the vendor.

- iii. To hire experienced contractor

In case 4, two contractors had been hired one after the other to take over the development work.

To resolve management issues

- 1) To get better management support

- i. To get project funded properly

In some cases, the projects were fully funded and assigned with all the necessary resource. On another case, as the project was not funded, there

wasn't much management support and dedicated resource, the project was largely abandoned in the end.

ii. To get business users involved

The involvement of business in the development of the cases varied a lot. In case 1, business had involved the most and the development had got the most support from management. In some other cases, there might be not much business involvement; the projects were not supported properly.

2) To manage the project properly

i. To assign project manager with responsibility and accountability

In case 1 and 2, there were dedicated project managers to ensure the delivery. On other cases such as case 4, the project was delayed for over 6 months initially without a dedicated project manager.

ii. To follow project management methods

In case 1, a trail of project management documents were found in the research, and the development was successful in the end. On other not so successful cases, many discussion records were found but crucial project management including progress checking and updating were missing.

To ensure business involvement

1) To involve them in early stage

i. In case 1 and 2, the business was involved from the beginning the analysis stage. Late business involvement like in case 4 would be more difficult.

2) To provide prototype and run workshop

i. In case 1, prototype and workshop were the main method to get business involved and to provide feedbacks.

To improve user acceptance/training

1) To included plan and resource for user acceptance and training

i. In case 1 and 2, user training was included in the project plan which had allowed for time and resource for it.

2) To make sure the requirements are clearly defined and agreed

i. In case 1 and 2, since the requirements were well defined and agreed by the business, it had make the acceptance and training much easier than otherwise.

- 3) To involve business users
 - i. In case 1 and 2, business rep had contributing a lot in creating training material and running training for users.

5.3.5 What would be the XML applications for these integrations?

Summary of XML applications in integration systems

From the case selection, the researcher had found XML application would be in three areas: data integration, functional integration and settings for the integration systems.

1) Data integration

XML web services were used for data integration in many cases for the following reasons:

- i. Loose coupling

In most cases, XML web services were adopted because they did not incur problems from tight coupling integration through low level database object linkage. XML web services would be easily configured and shared by multiple systems.

- ii. Data process

XML web service was adopted in case 2 for processing data because it was more flexible than the previous method.

2) Functional integration

The reasons for using XML in this area were:

- i. Third party software factor

In case 1, XML was already used as data format by the third party sub-system, the integration system had to work with it.

- ii. Multiple platforms

In case 2, XML web service was adopted to provide integration functions to sub-systems with different .Net versions.

3) System settings

XML was found used for system settings in the follow areas:

- i. Web based systems

All web-based integration systems in the case selection use XML files for the system configurations.

ii. Integration Configuration

In case 1, extra XML files were used to store setting for the integration functions.

Summary of technical CSFs for the XML applications

The research had found the technical critical success factors for XML application in the case selection were: XML development tools, development skills and development methodology.

1) XML development tools

It was found to be the most important factor in many cases. Only with the availability of these tools in the development environment, could new integration such as the GIS Viewer in case 1 have been developed. In other cases, because of the lack of the tools, successful integrations had not be developed.

2) Development Skills

In case 1, successful integration was developed by a more experienced developer. In another case 4, the lack of development skills led to the faulty initial design without proper integration technique such as XML web service.

3) Development Methodology

In one case, successful integration was developed by following development methodology. In case 3, ad-hoc design and development of XML web service had led to very limited success of the system.

Chapter 6 Conclusions and Discussions

6.1 Conclusions

6.1.1 Case Study Conclusions

From the case study, the following were concluded:

- 1) ***The core issues for developing system integration would be from the requirements, the solution design and the integration development.***

Although all the systems and their integrations in the cases were somehow unique, however common patterns had been identified in the above areas from the research. Those issues were crucial for the integration and they were the major factors for the project being successful or not as found in the case study.

- 2) ***There would be effective measures to resolve those issues.***

As found in the case study, integration requirements must be kept basic and clear. Using prototype and getting user involvement were found effective for achieving that. Solution design was very important, while developing proof of concept and adopting suitable architecture were found to be effective for getting good solution design. The development of the integration solutions would be complicate. While it was missing in those not so successful projects, adopting effective development methods such as prototyping and iterative development were found effective for those successful projects in the research.

- 3) ***Technical factors would be more critical for small to medium size integration development.***

Although management support and project management were important factors for all integration projects, knowledge of the existing systems and adequate integration design were found to be the crucial factors in all cases, they would have profound impact on the outcome of the projects.

- 4) ***XML application was important for the integrations.***

XML applications were found to be the essential integration technique in some cases; critical success factors for those XML applications include suitable development tools, development skills and methodology.

6.1.2 Enterprise Application Integration Models

A set of models for enterprise application integration had been derived from the cases as common patterns identified in the research.

1. Integration Requirement Model

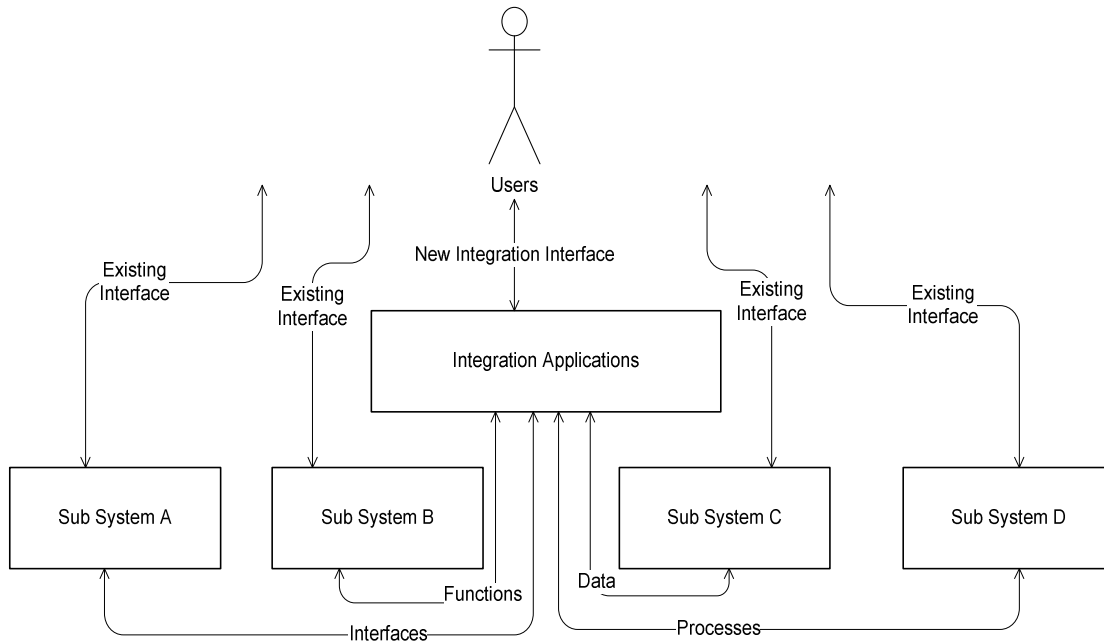


Figure 6. 1 Integration Requirement Model

The integration requirement model has illustrated the following:

- 1) Unlike other integration approaches such as ERP (SAP) systems, integration systems in the case study were built on top of the existing systems and not to replace them. Existing systems will still in use after the integrations were completed. This rule applied to all cases in the research.
- 2) Most integration systems had their own user interface. This applied to all cases in the research, except in case 3 which was focused on data sharing in most stages, but new user interfaces were proposed (for the PerOrg project) in one of the multiple phases.
- 3) Integrations would provide a new set of features by combining the power of existing business systems. Integration systems would combine and interact with data, functions, interfaces and processes from all the underlying systems, the labelling of the sub systems and specific integration requirement in the diagram are for illustration only.

- 4) The model illustrated why clear and basic requirement was crucial for the success of the integration. As integration was to provide new features including new user interface, the requirements must be clearly defined, as in case 1 and 2. Integration was not to replace or take over from the existing systems, so its function should be basic, otherwise the project scope would be hard to control and the development would be in jeopardy, such as the project in case 4.

2. Integration Solution Design Model

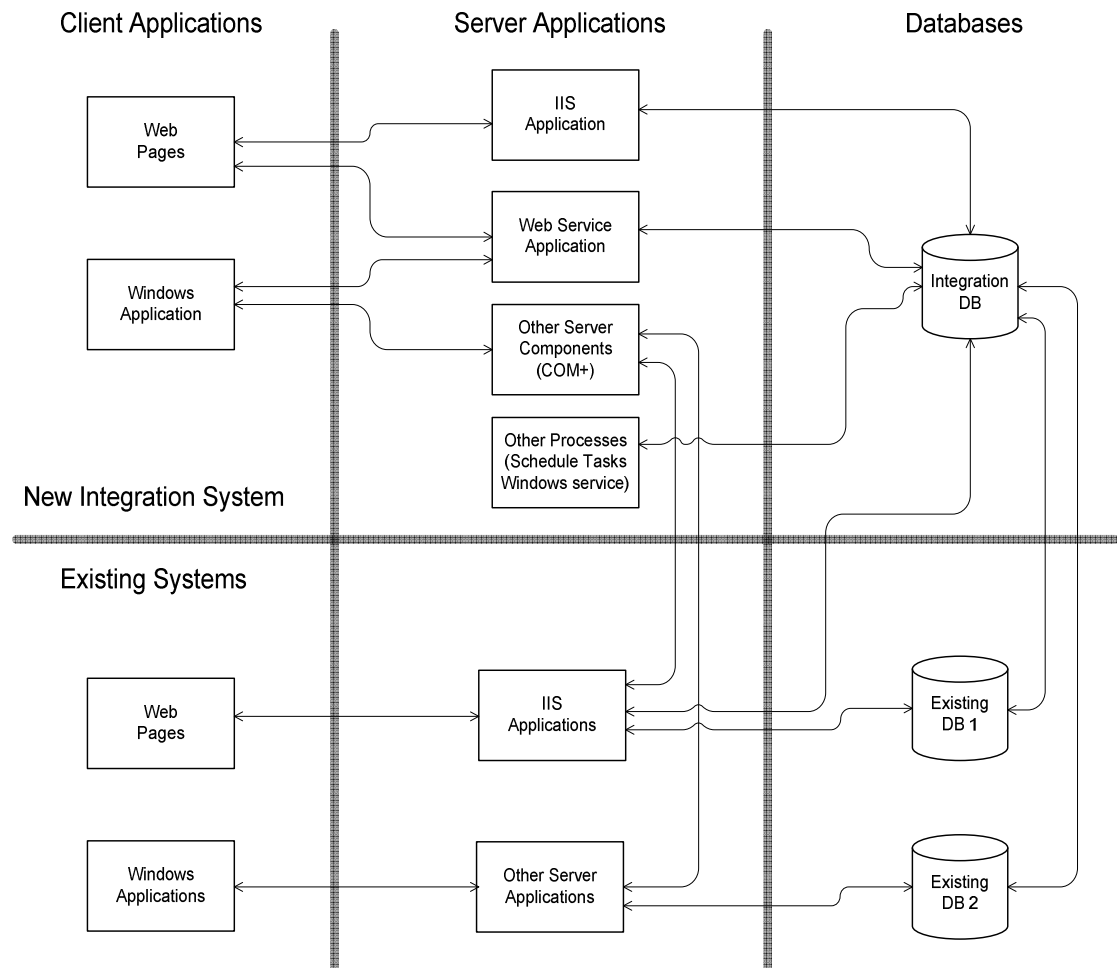


Figure 6. 2 Integration Solution Design Model

The integration solution design model has illustrated the following:

- 1) Integration solutions would follow the pattern of other existing systems, but they would be a lot more complicated than a single system. Integration systems had to connect to multiple sub-systems in multiple levels as illustrated in the model. There would be more integration applications (components) running on the server side than existing systems, as found in all cases in the research.

- 2) Server applications for the integrations would be very complex and of various types. IIS web sites and web service were the most common ones found in the research. Some of those other types found in the research included stand alone application running as scheduled task, such as DOCgdi for updating the GIS databases. There are also COM+ application for communicating to the DM server and Windows service for regular GIS status update for Permissions and land register.
- 3) The inter-relationships and dependencies among the integration and existing systems would be very complex. Each of the integration component including those in client side, server side and database would interact with other components of the existing systems. The above model only shows a small sub-set of those relationships, which were the most common ones. Many unusual ones are not shown, for example, in case 2, the DM integration component on the client side actually activates the DM client side component for creating a new document from the client PC.
- 4) XML data format and XML applications such as web service would be best adopted for developing the interactions and data exchange among all the components. Although it would be quicker and easier by using other means in some situations, such as those cross databases views in the early development stages in GIS integration, however, the benefit of XML and loose coupling integration would be for longer term and more robustness of the systems. The development of GIS Viewer in case 1 and web service in case 2 were the good examples.

3. Integration Development Model

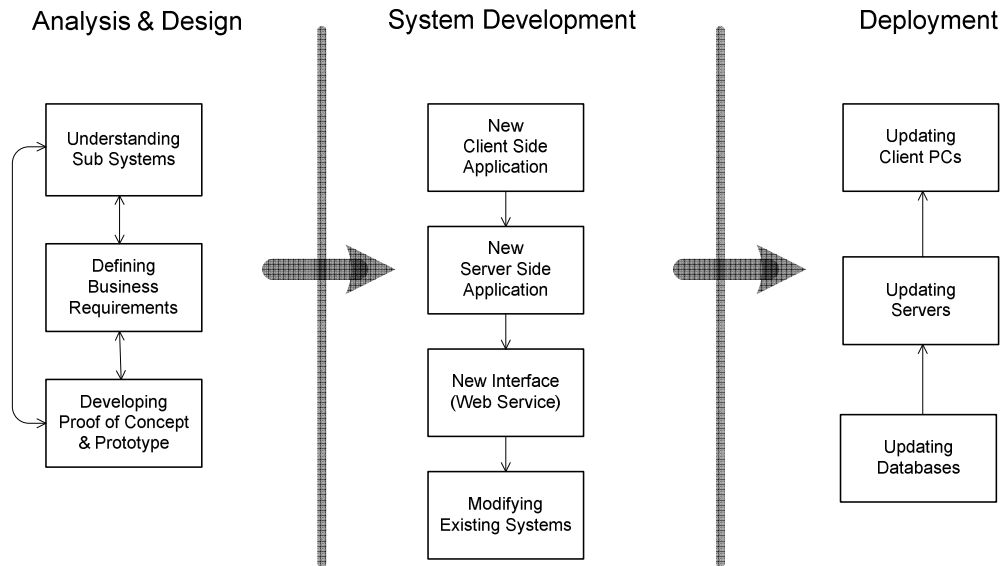


Figure 6. 3 Integration Development Model

The integration development model has illustrated the following:

- 1) The development of the system integrations in the research would have followed similar phases of traditional (water-flow) system development model, including analysis & design, development and deployment, but it had been modified to include special system integration elements such as the iterative approach in analysis design.
- 2) With integration development, the analysis & design phase would be most important and challenging. It was successfully done in case 1 and case 2 partially by going through small iterative circles of learning the sub systems, defining the requirements, developing proof of concept and prototype. The use of methods such as proof of concept and prototyping were crucial, which was consistent with findings in the literature review. The phase has to be done as long as it would take. In case 1, the analysis and design phase was actually longer than the development, in the end, the solution was a successful one. In case 2, the pushing for delivery in the initial development had left issues for fixing later. The research had found no such attempt in case 3 and case 4, and both of them had unsatisfactory results. Especially in case 4, the project was started almost right into the development phase without initial analysis as pointed out in the final report.
- 3) Integrations would involve a few more development tasks than usual business systems as found in the research. The diagram shows a top-down approach of the

main system development phase, which was found in all cases in the research, but as the literature had found, the development would also be of bottom-up approach. No iteration development approach was found in the main development in all cases, which would be due to the relatively small size of the projects.

- 4) The deployment of integrations would have involved a few tasks. The diagram shows a bottom-up (database, server, and client) approach, which was found in all cases in the research. Such approach would ensure each step would be testable and minimize impact to existing systems.

4. Critical Success Factor (CSF) Model

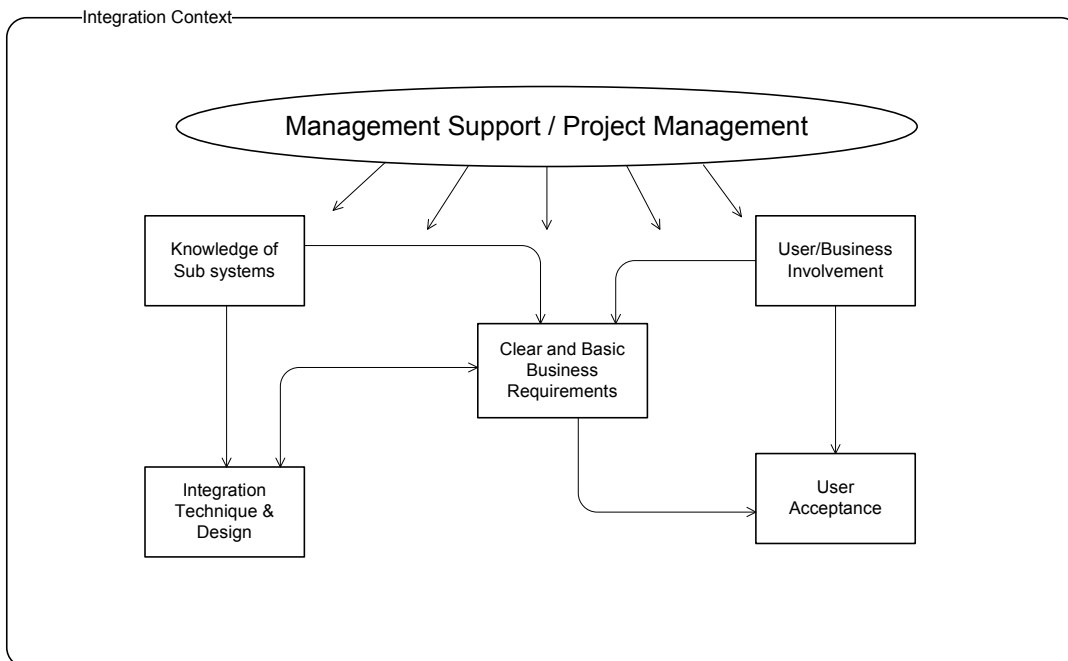


Figure 6. 4 Critical Success Factor (CSF) Model

The critical success factor model has illustrated the following:

- 1) Critical success factors would interact with each other during the development life cycle. The Management Support/Project Management factor would affect everything within the entire context for the integration development. It had affected all the other factors, as it was found in all the projects in the research. Although project management was singled out only in case 4 as the top reason for the project's unsuccessful result, it was identified as a contributor for the success of cases 1 and case 2.
- 2) Most of the factor would affect each other as implied in the diagram. For example, good knowledge of sub-system would help to define business requirements and

ensure workable integration design; clearly defined business requirement would help in finding suitable integration technique and ensuring user acceptance; and etc.

- 3) From the analysis of the cases, business requirement was found to be the most important factor. In both case 1 and 2, requirements were clearly defined and had ensured the development, delivery and acceptance of the integrations; while in case 3 and case 4, the integration projects were not completed, one of the main reasons was there was not clearly defined and agreed user requirements.
- 4) Knowledge of sub-systems and integration techniques were found to be the most important technical factors in all cases. Both of them would affect the business requirement factor. The knowledge of sub-systems was important factor for choosing the integration techniques as found all cases.
- 5) The research had only summarized a small set of CSF factors, which were of high level. An in-depth study of CSF was out of the scope of this research.

6.1.3 Discussions

1. Research on common system integration development method is much needed.

The system integrations included in this research were selected from those common systems in the agency. The researcher had found the challenges arising from them had not been addressed sufficiently either from the literature reviewed or in the commercial world. More research in this area is urgently needed. The reasons would include:

- 1) *The cases would not be effectively covered by any of the current system integration model, such as data integration and workflow integration.*

As found in the research, integration requirements would come from many directions. In some cases, they would be just a combined function like creating a document automatically in case 2. In other cases, more complex processes would be involved, like case 4. Even in complex case, a common integration model like workflow would not be expected or be necessary. As by adopting any of the models, business process would have to change accordingly. In all the cases of this research, the purposes of the integration were either to combine features of multiple systems, or to share data, or to automate processes. None of the integration was expected to significantly impact on the current business processes.

- 2) ***They were small and various integrations and not covered by general EAI theories either.***

The integrations in this research would be regarded as some sorts of enterprise application integration (EAI) as they were among applications within the agency. But because of being relatively small in size such as case 2, or specialist software involved such as case 1, no general EAI framework such as an EAI adaptor could be found to be applicable for these cases.

- 3) ***Many more small-scope EAI would likely to happen than large EAI in an organization.***

As the majority of IT work is done within the organization, some kind of integration among existing systems would be unavoidable. As found in this research, many of these integrations would be small in size but vary in requirements. Also because the agency had standardized the majority of its software in one platform, such integration would more likely to happen than otherwise.

2. Integration development would be more complicate than other system development.

The research did found something consistent with what was found from the literature review, which must be seriously considered for future development:

- 1) ***Complexity of integration***

System integrations are generally regarded more complex than normal system development. They normally involve more components and require upfront planning. The research had found some of the cases involved a number of sub-systems, such as in case 4. The software company had grossly underestimated the complexity of the integration and failed to plan and design a workable solution initially. All of the other cases contained components or interfaces that somehow had caused issues because of complicate integration functions.

- 2) ***Development Method***

Regardless of the scope of the projects, some form of development methods should be adopted and followed. Basic methods like prototyping were found very effective, including case 1 and 2, which was consistent to the review findings. In some other cases, development methods were not followed well, which would have caused the unsatisfactory results.

3) ***Basic set of critical success factors (CSF)***

The research have found a basic set of CSFs from the research, they were clear and basic business requirement, knowledge of the sub-systems, integration technique and management support & project management. All of these factors were also reflected in the paper by Lam (2005) for EAI.

4) ***XML applications***

Many of the papers in the literature review argued XML web service would be extensively used for the integrations. The research has identified XML applications in all the cases. The following are the benefits of XML found in the case study, which also found in book by Mike Clark, Peter Fletcher, J. Jeffrey Hanson, Romin Irani, Mark Waterhouse, Jorgen Thelin (2002).

i. ***Simplicity and Effectiveness***

The research found that XML web service was integrated part of the standard development platform for the agency. Using XML or developing XML web service would be simply and effective. The research had found in most of the cases, XML or XML web service were adopted. In case 1, XML was already used by the third party software as input-output data format, it was simple and effective for the integration system to use. XML web services were also found to be simply and effective interface, such as in case 4 for interfacing the HRIS system.

ii. ***Platform Independence***

The research found that XML web services were used in some cases for interface multiple systems regardless of their platform. In case 3, the XML web service was planned to be used by many business systems from all kind of source. In case 2, an XML web service was adopted to provide integration functions to systems from different .net version.

3. Using a model based framework for small to medium size integration development should be considered.

As found in the research, the nature of the integrations in the agency varied from case to case, it was difficult to establish an abstract model from the case selection.

However, a set of basic models had been derived from the research and they would be used for guiding the development for future integrations. A framework has been derived and described as following:

The Model Based Enterprise Application Integration Development (MB-EAID) Framework

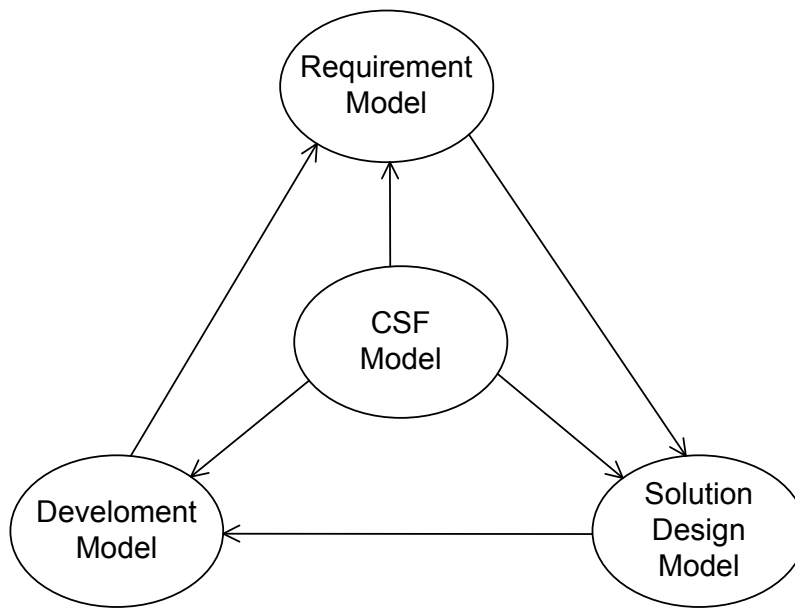


Figure 6. 5 Model Based Enterprise Integration Development (MB – EAID) Framework

Descriptions for the MB-EAID framework:

- 1) The requirement model would be used for defining integration requirements. It is the first model to be used within the framework. Clear and basic integration requirements are crucial for the success of the integration.
- 2) The solution design model would then be used for designing integration components and identifying programming interfaces among the new integration components and existing ones.
- 3) The integration development model would be used to guide the development process. The analysis and design is one of the most important phases. A small iterative approach in this phase would be useful and effective. The rest of development would also be handled accordingly, either by top-down or bottom-up approach. The delivery from the development phase would then be validated by users according to the business requirements.
- 4) The critical success model would be used to govern all the processes within the framework and ensure the success of the overall integration development. Within the CSF model, management factor would have overall impact on all other factors. Requirements would be the central of the issues, as it would affect or be affected

by many other factors. Knowledge of sub-systems and integration technique would be the most important technical factors that would affect the solution design and development.

4. New user interfaces would be important in system integration and the CSFs would vary depending on the scope of the project.

There were some findings from the research that were different from the literature review in the following areas, which must be noticed for future project.

1) *Middleware or User Interface*

Many of the EAI theories argued for middleware being the most important integration component. Many of EAI frameworks were found in the literature review focused on the integration middleware (adapter). However, this research found integrations in all cases contained user interface, some were the core parts of the integrations, such as case 1, 2 and 4.

2) *Top Critical Success Factors*

In Lam's (2005) paper, he argued the top three CSFs for EAI were top management support, overall integration strategy and realistic project plans & schedule. The research found the top three CSFs in the case selection were business requirement, knowledge of the sub-systems, integration technique. Management issue was found to be the top factor only in case 4, which was relatively a large project. The scope of project would have impacted on the CSFs. Technical factors were found likely to be more important in small projects (e.g. case 2) than in large ones (e.g. case 4).

5. XML application should be adopted and many factors need to be considered.

- 1) XML applications including XML configuration, XML data format and XML web service should be seriously considered, especially in the programming interfaces identified in the architecture model. As they are effective and easy to build and maintain. They are platform independent and open to future change and new integrations. After all, XML would be the key for building loose coupling and robust integration solutions.
- 2) XML application development would be affected by a few factors, including development tools, developer skills and development method. XML applications

would be the keys for successful integrations, but they would require advanced techniques, simple XML web service may not be enough, as it was found in case 3.

6.2 Research Contributions

The contribution of the research to the study of the discipline would come from a number of areas, including the literature review, the research method and the research conclusion.

6.2.1 The Literature Review

In the literature review, the researcher had done extensive review covering IS development, system integration and XML technologies. It would be useful for fellow practitioners or academic who also have interest in those areas. The review also had covered some of the IS research methods, with particular interest in case study.

6.2.2 The Research Method

This research has executed a case study in a custom designed approach. Although there is hardly any formal pattern for case study as cases tend to be different from each other. The case study method in this research had some special aspects, as described in the following:

- 1) The case study had used documentation including source code extensively.
Documentations were the primary source for this research for a number of reasons, including:
 - i. They were stable, unobtrusive, exact and of broad coverage.
 - ii. They would be authored by multiple people for multiple purposes; hence, they would be used for crosschecking or forming chain of evidence.
 - iii. They were plentiful and accessible.
- 2) The researcher has extensive context knowledge on the cases.
Since the researcher had worked in the agency for many years and also had extensive experience in IT development, the benefits for the research included:
 - i. The researcher would understand both the business and technology issues well.
 - ii. The data collection for the research would be more efficient because the researcher knew where to find them.
 - iii. The researcher would have access to data source that would normally be not available.

- 3) The research had used multiple analysis techniques.

As there was a large quantity of data collected for the research, the researcher was able to use multiple analysis techniques including triangulation, chain of evidence, root cause analysis, counterfactual comparison and pattern matching.

If the conclusions of the research are proven valid and the research method is proven effective, there will be benefits for future research, which may include:

- 1) Research can be done effectively using solid documentations without overly depending on other traditional sources, such as interviews.

Doing research with documentation would be more effective and efficient than using other source, such as interviews. Interviews would be useful but they require interviewees to be available. Interviewees were not available in this research as the cases were in the past and most involved staff had left the agency. By using documentations left behind including system source code, the researcher was able to carry out the case study and derived the conclusions.

- 2) Practitioners in the field can contribute more in research to the area they work in. Since the researcher has worked for the agency and has strong interest in the areas, he would be able to contribute to the study area more if effective research method would be found and used. Research methods, which involve identifying and going into external organizations, preparing and conducting interviews and other data collection procedures would be time consuming and still produce unreliable results. On the other hand, practitioners would have plenty of experience in the actual systems and access to large quantity of data source which is not usually available to external researchers; they would have special advantages on carried out the research as long as strict validity measures were applied. The researcher believes other practitioners would more likely to contribute to the topic in their work area, if they know how to do research using effective methods, such as the one adopted in this research.

6.2.3 The Research Conclusion

The conclusions were deduced from the case study research with large amount of data, analysis and quality assurance measures. The target area is part of the day-to-day IT works within a large government agency regarding system integrations. Although there are a lot of theories and work done before, but most of them are either for much larger

scale of system integrations that would be comparable to a total ERP approach, or for particular kind of integration such as data sharing and B2B system. Those theories and frameworks are not very suitable in the situations found in the research, mainly because most of IT works happen within the organization are in small scales with specific requirements. This research had pay attentions to these areas. Some cases selected in the research would be relatively small, but they would happen more often than large-scale projects, and they would be complex too. All these integration systems in the research were important for the agency, but would not have been covered in other research.

The conclusions of the research would be important for organization and developers who have engaged or planed to engage in information systems development that may integrate or simply interface with multiple systems. The conclusions had included a set of integration models and a development framework. They would be used as guidelines for future integration systems or starting points for further research. They are still not conclusive or readily generalisable in the end of this research, as the environment of IS development would not be simply duplicated and more research would be needed. The conclusions should be used as reference only and other factors such as the development methodologies and the project management should be seriously considered.

6.3 Research Limitations

There are some limitations in this research that must be disclosed and to be taken into account for further investigation and future research.

- 1) The research was focused on technical issues although some business and management topics were also covered. In many situations that identified within this research, management and human behaviours issues could have very important impact on IT development including system integrations.
- 2) The research was carried out within a single organization, which has imposed standards and constraints on all of the existing systems and new integrations, i.e. systems were all based on Microsoft technologies. They were the prerequisites of the integrations and also limitations of the research conclusions. For generalization

of the findings, the case selection should have been representative of a large area, as argued by John Gerring (2007).

- 3) The research objectives covered integration and XML, which are big topics by them self. The research would not have done in-depth analysis on all issues identified in the research, such as critical success factors for integration development.
- 4) For data collection, the research relied heavily on the context knowledge of the researcher and the project documentation that normally only accessible by staff member. The understanding of the source code required in-depth technical knowledge that the researcher happened to have. Such approach may not be easily repeated by other researchers.

6.4 Future Research

The followings can be considered for future research and study; many of them were identified from the review of the draft report by fellow IT professionals:

- 1) ***To carry out the research in another agency.***

As argued by John Gerring (2007), the case selection must be representative of large area for generalization. In order for getting more conclusive result from the research, it needs to expand the case study to include more organizations.

- 2) ***To expand the research to cover more business and management topics.***

In order to grain good insights and a complete picture of the research topic, the research needs to cover all important areas including human behaviours and management issues. As the nature of information systems, they are heavily affected by those factors unlike other topics of pure technology.

- 3) ***To concentrate on smaller area, such as CSF for integration development or XML application for integration.***

As pointed out as a limitation, this research was difficult because of its topics were too big to do any in-depth analysis or to do advanced exploration. A future research should be considered for concentrating on a certain area that allow the research to gain in-depth insights and explore advanced techniques, such as adopting advanced XML technologies, e.g. XQuery.

- 4) ***To develop or validate theories or development models and techniques for small to medium system integrations.***

The results of this research are not conclusive. A future research would be considered to develop theories for this area, which would be tested or proven using other real case scenarios. A research would generate much more value if it could product results which would be applied in future.

5) ***Service Oriented Architecture (SOA).***

Many web services were identified in the research, but most of them were for providing particular functions. SOA was identified in the literature review as a major trend for integration design including the use of web service. A new concept of Enterprise Service Bus (ESB) has also emerged and identified in the literature review. It is a further expansion of SOA. A future research with focus on SOA or EBS would be potentially interesting and challenging.

6) ***Workflow Integration.***

The case 4 in the research was identified to have a few workflow elements; the project was proven to be a challenge and given a few lessons for the agency to learn. In the literature, workflow was identified as a major challenge for system integration, and many workflow frameworks were identified. However, in a project like case 4, how would such a framework be applied and whether or not the cost and benefit for applying any workflow framework would be justified were questionable. A future research for finding a light weight and flexible workflow framework would be very beneficial for organizations with similar projects.

Reference

- Anda, Bente; Hansen, Kai (2006), A case study on the application of UML in legacy development, Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering
- Arion, Andrei; Bonifati, Angela; Manolescu, Ioana; Pugliese, Andrea (2007) XQueC: A Query-Conscious Compressed XML Database, ACM Transactions on Internet Technology, Vol. 7, No. 2, Article 10
- Arthorne, N. ; Esfandiari, B. (2006) Peer-to-peer data integration with distributed bridges, October 2006, Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research
- Aslan, Goksel; McLeod, Dennis (1999) Semantic heterogeneity resolution in federated databases by metadata implantation and stepwise evolution, The VLDB Journal (1999) 8: 120–132
- Baghdadi, Youcef (2005), A Web Services-Based Business Interactions Manager to Support Electronic Commerce Applications, ICEC'05, August 15–17, 2005, Xi'an, China.
- Bao, Yimin; Horowitz, Ellis (1996), Integrating through user interface: a flexible integration framework for third-party software, Proceedings of 20th International Computer Software and Applications Conference, 21-23 Aug 1996 Page(s):336 – 342
- Benbasat, I., Goldstein, D.K. and Mead, M. (1987) "The Case Research Strategy in Studies of Information Systems," MIS Quarterly (11:3) 1987, pp. 369-386.
- Berntsson-Svensson, Richard; Aurum, Aybuke (2006), Successful Software Project and Products: An Empirical Investigation, ISESE'06, September 21–22, 2006, Rio de Janeiro, Brazil.
- Bertolino, Antonia; Gao, Jinghua; Marchetti, Eda; Polini, Andrea (2007) TAXI - A Tool for XML-Based Testing, 29th International Conference on Software Engineering (ICSE'07 Companion)
- Bhatti, Rafae; Ghafoor, Arif; Bertino, Elisa; Joshi, James B. D. (2005) X-GTRBAC: An XML-Based Policy Specification Framework and Architecture for Enterprise-Wide Access Control, ACM Transactions on Information and System Security, Vol. 8, No. 2, May 2005, Pages 187–227
- Bichler, M; Segev, A; Zhao, JL.(1998), "Component-based Ecommerce: assessment of current practices and future directions". ACM SIGMOD 98 Rec 27(4): 7-14
- Biehl, Markus (2007), Succes Factors for Implementing Global Information Systems, Communication of the ACM, January 2007/Vol. 50, No. 1
- Bingi, P.; Sharma, M.K.; Godla, J.K. (1999), Critical issues affecting an ERP implementation. Information Systems Management 16, 3 (Summer 1999), 7–14.
- Bryan, G.M.; Curry, J.M.; McGregor, C.; Holdsworth, D.; Sharply, R. (2002) Using XML to facilitate information management across multiple local government agencies, Proceedings of the 35th Hawaii International Conference on System Sciences
- Bygstad, Bendik; Nielsen, Peter Axel; Munkvold, Bjørn Erik (2005), Four Integration Patterns: IS Development as Stepwise Adaptation of Technology and Organization, Proceedings of ECIS2005, Regensburg, Germany, May.2005
- Calvanese, Diego; De Giacomo, Giuseppe; Lenzerini, Maurizio; Rosati, Riccardo (2004) Logical Foundations of PeerTo
- Chua, Cecil Eng H; Chiang, Roger H. L.; Lim, Ee-Peng (2003) Instance-based attribute identification in database integration, October 2003, The VLDB Journal — The International Journal on Very Large Data Bases, Volume 12 Issue 3
- Clark, Mike; Fletcher, Peter; Hanson, J. Jeffrey; Irani, Romin; Waterhouse, Mark; Thelin, Jorgen (2002), Web Services Business Strategies and Architectures, Wrox Press (August 2002)
- Cohen D, Crabtree B. "Qualitative Research Guidelines Project." July 2006. Retrieved June 10, 2009, from <http://www.qualres.org/HomeMemb-3696.html>
- Crinnion, John (1992), The Evolutionary Development of Business systems, A paper presented at the IEE Colloquium on 'Software Prototyping and Evolutionary
- Damiani, Ernesto; Vimercati, Sabrina De Capitani di; Samarati, Pierangela (2002) Towards Securing XML Web Services, ACM Workshop on XML Security, November 22, 2002, Fairfax VA, USA.
- Damodaran, Suresh (2004), B2B integration over the Internet with XML: RosettaNet successes and challenges, Proceedings of the 13th international World Wide Web conference on Alternate track (2004) Pages: 188 – 195
- De Villiers, M. R. (2005) Three approaches as pillars for interpretive information systems research: development research, action research and grounded theory, Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries
- Degwekar, Seema; DePree, Jeff; Beck, Howard; Thomas, Carla S.; Su, Stanley Y. W. (2007) Event-triggered Data and Knowledge Sharing among Collaborating Government Organizations, The Proceedings of the 8th Annual International Digital Government Research Conference
- Denzin, NK. (1978). Sociological Methods. New York: McGraw-Hill.
- Emmerich, Wolfgang; Ellmer, Ernst; Fieglein, Henry (2001) TIGRA — an architectural style for enterprise application integration Source International Conference on Software Engineering archive, Proceedings of the 23rd International Conference on Software Engineering table of contents, 2001

Reference

- Gabillon, Alban (2004) An Authorization Model for XML DataBases, Conference: ACM Workshop on Secure Web Services, DatePlace: October 29, 2004, Fairfax VA, USA.
- Gerring, John (2007) Case Study Research: principles and practices , Cambridge University Press (2007)
- Getta, Janusz R. (2006), Optimization of Online Data Integration, School of Information Technology and Computer Science, University of Wollongong, Wollongong, NSW 2522, Australia
- Glass, Robert L.; Ramesh, V.; Vessey, Iris (2004) An analysis of research in computing disciplines, June 2004 Communications of the ACM, Volume 47 Issue 6
- Goede, Roelien; de Villiers, Carina (2003) The Applicability of Grounded Theory as Research Methodology in studies on the use of Methodologies in IS Practices, September 2003 SAICSIT '03: Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology
- Gorton, Ian; Thurman, Dave; Thomson, Judi (2003), Next Generation Application Integration: Challenges and New Approaches, Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC'03)
- Gou,Gang; Chirkova, Rada (2007) Efficiently Querying Large XML Data Repositories: A Survey, IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 10, October 2007
- Gudivada, V.N.; Nandigam, J. (2005), Enterprise application integration using extensible Web services, Proceedings. 2005 IEEE International Conference on Web Service
- Guha, Sudipto; Jagadish, H. V.; Koudas, Nick; Srivastava, Divesh; Yu, Ting (2006) Integrating XML Data Sources Using Approximate Joins, March 2006 ACM Transactions on Database Systems (TODS), Volume 31 Issue 1
- Halevy, Alon Y. (2001), Answering queries using views: A survey, The VLDB Journal, Issue Volume 10, Number 4 / December, 2001
- Halevy, Alon; Rajaraman, Anand; Ordille, Joann (2006) Data integration: the teenage years, VLDB '06: Proceedings of the 32nd international conference on Very large data bases
- Halpin, Harry; Thompson, Henry S. (2006) One Document to Bind Them: Combining XML, Web Services, and the Semantic Web, WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
- Hazzan, Orit; Dubinsky, Yael; Eidelman, Larisa; Sakhnini, Victoria; Teif, Mariana (2006) Qualitative research in computer science education, ACM SIGCSE Bulletin archive, Volume 38 , Issue 1 (March 2006)
- He, Wen; Lu,Peng; Zhou,Chao; Lin, Xuelian (2006) Web Based E-Government Information Sharing, 2006 IEEE International Conference on System Information Reuse and Integration, Sept. 2006 Page(s): 63 - 68
- Head, Michael R.; van Engelen,Robert; Zhang, Wei (2006) Benchmarking XML Processors for Applications in Grid Web Services, Supercomputing, 2006. SC '06. Proceedings of the ACM/IEEE SC 2006 Conference
- Helander, Johannes (2004) Deeply Embedded XML Communication - Towards an Interoperable and Seamless World EMSOFT'05, September 19–22, 2005, Jersey City, New Jersey, USA.
- Holz, Hilary J.; Applin,Anne; Haberman,Bruria; Joyce,Donald; Purchase,Helen; Reed, Catherine (2006) Research methods in computing: what are they, and how should we teach them? June 2006 ITiCSE-WGR '06: Working group reports on ITiCSE on Innovation and technology in computer science education
- Ives, Zachary G.; Halevy,A. Y.; Weld, D. S. (2002) An XML query engine for network-bound data, December 2002 The VLDB Journal — The International Journal on Very Large Data Bases, Volume 11 Issue 4
- Johnstone, Lynne (2007), Weighing Up Triangulating and Contradictory Evidence in Mixed Methods Organisational Research, International Journal of Multiple research approaches Volume 1 Issue 1 - 2007
- Kajan, Ejub; Stoimenov, Leonid (2005) Toward an ontology-driven architectural framework for B2B, December 2005, Communications of the ACM, Volume 48 Issue 12
- Kangasharju, Jaakko; Tarkoma, Sasu (2007) Benefits of Alternate XML Serialization Formats in Scientific Computing SOCP'07, June 26, 2007, Monterey, California, USA.
- Khoubati, Khalil; Themistocleous, Marinos; Irani, Zahir (2006), Evaluating the Adoption of Enterprise Application Integration in Health-Care Organizations, Journal of Management Information Systems / Spring 2006, Vol. 22, No. 4, pp. 69–108.
- Kim, Hee-Woong; Pan, Shan L (2006), Towards a Process Model of Information Systems Implementation: The Case of Customer Relationship Management CRM), The DATA BASE for Advances in Information Systems - Winter 2006 (Vol. 37, No. 1)
- Klein, H.K; Myers, M.D. 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. MIS Quarterly,
- Kostoulas, Margaret G.; Matsa,Morris; Mendelsohn,Noah; Perkins,Eric; Heifets,Abraham; Mercaldi, Martha (2006) XML Screamer An Integrated Approach to High Performance XML Parsing, Validation and Deserialization, May 2006 WWW '06: Proceedings of the 15th international conference on World Wide Web
- Kurita, Hiroto; Hatano,Kenji; Miyazaki, Jun; Uemura, Shunsuke (2007) Efficient Query Processing for Large XML Data in Distributed Environments, 21st International Conference on Advanced Networking and Applications(AINA'07)
- Kwa, Myungjae; Han, Dongsoo; Shim, Jaeyong (2002), A Framework Supporting Dynamic Workflow Interoperation and Enterprise Application Integration, Proceedings of the 35th Hawaii International Conference on System Sciences - 2002
- Lam, Wing (2005) Investigating success factors in enterprise application integration: a case-driven analysis, European Journal of Information Systems (2005) 14, 175–187

- Lee, Jinyoul; Siau, Keng; Hong, Soongoo (2003), Enterprise Integration with ERP and EAI, February 2003/Vol. 46, No. 2 Communication of the ACM
- Lee, Mong Li; Yang, Liang Huai; Hsu, Wynne; Yang, Xia (2002) XClust: Clustering XML Schemas for Effective Integration, November 2002 Proceedings of the eleventh international conference on Information and knowledge management CIKM '02
- Lenzerini, Maurizio (2002) Data Integration: A Theoretical Perspective, Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Madison, Wisconsin
- Li, Lei; Singh, Roop G.; Zheng, Guangzhi Art Vandenberg, Vijay Vaishnavi, Sham avathe (2005) A methodology for semantic integration of metadata in bioinformatics data sources, ACM-SE 43: Proceedings of the 43rd annual Southeast regional conference - Volume 1
- Lóscio, Bernadette Farias; Costa, Thiago; Salgado, Ana Carolina (2006) Query reformulation for an XML-based Data Integration System, April 2006 Proceedings of the 2006 ACM symposium on Applied computing SAC '06
- Lu, Eric Jui-Lin; Wu, Bo-Chan; Chuang, Po-Yun (2003) An empirical study of XML data management in business information systems, The Journal of Systems and Software 79 (2006) 984–1000
- Lu, Zhongyu (2004) A Survey of XML Applications on Science and Technology, International Journal of Software Engineering and Knowledge Engineering, Vol. 15, No. 1
- Maheshwari, Piyush (2003), Enterprise Application Integration using a Component-based Architecture, Proceedings of the 27th Annual International Computer Software and Applications Conference
- Matsunaga, Andréa; Tsugawa, Maurício; Fortes, José. A. B. (2007) Integration of Text-based Applications into Service-Oriented Architectures for Transnational Digital Government, The Proceedings of the 8th Annual International Digital Government Research Conference
- Medjahed, B.; Benatallah, B.; Bouguettaya, A.; Ngu, A. H. H.; Elmagarmid, A. K. (2003) Business-to-business interactions: issues and enabling technologies, The VLDB Journal (2003) 12: 59–85
- Medjahed, Brahim; Benatallah, Boualem; Bouguettaya, Athman; Ngu, Anne H. H.; Elmagarmid, Ahmed K. (2003), Business-To-Business Interactions Issues, The VLDB Journal (2003) 12: 59–85
- Milo, Tova; Abiteboul, Serge; Amann, Bernd; Benjelloun, Omar; Ngoc, Fred Dang (2005) Exchanging Intensional XML Data, March 2005 ACM Transactions on Database Systems (TODS), Volume 30 Issue 1
- Myers, Michael D. (1997) Qualitative Research in Information Systems, MISQ Discovery, June 1997
- Naidu, Pavithra G.; Palakal, Mathew J.; Hartanto, Shielly (2007) On-the-fly data integration models for biological databases, March 2007 SAC '07: Proceedings of the 2007 ACM symposium on Applied computing
- Nilsson, Erik G.; Nordhagen, Else K.; Oftedal, Gro (1990) ASPECTS OF SYSTEMS INTEGRATION, Center for Industrial Research (SI), Forskningsveien 1, P.O. Box 124, Blindern, N-0314 Oslo 3, Norway
- Nuldén, U. (1996). Escalation in IT Projects: Can We Afford to Quit or do We Have to Continue. Information Systems Conference of New Zealand, Palmerston North, New Zealand, IEEE Computer Society Press, p.136-142.
- Patton, MQ. (1999). "Enhancing the quality and credibility of qualitative analysis." HSR: Health Services Research. 34 (5) Part II. pp. 1189-1208.
- Pendyala, Vishnu S.; Shim, Simon S.Y.; Gao, Jerry Z. (2003), An XML Based Framework for Enterprise Application Integration, Proceedings of the IEEE International Conference on E-Commerce (CEC'03)
- Pentakalos, Odysseas (2008), Proof-of-Concept Design, Microsoft MSDN Architecture Center
- Perry, Dewayne E.; Sim, Susan Elliott; Easterbrook, Steve (2006), Case Studies for Software Engineers, Software Engineering Workshop - Tutorial Notes, 2005. 29th Annual IEEE/NASA, Volume , Issue , 3 & 8 April 2005 Page(s): 96 – 159
- Perry, Ivor (2003) Workflow by the Back Door? Using XML Systems in Health Service Processes, and Changing the System, Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)
- Pierce, Marlon; Fox, Geoffrey; Youn, Choonhan; Mock, Steve; Mueller, Kurt; Balsoy, Ozgur (2002) Interoperable Web Services for Computational Portals,
- Pottinger, Rachel, Bernstein, Philip A. (2008) Schema merging and mapping creation for relational sources, March 2008 EDBT '08: Proceedings of the 11th international conference on Extending database technology: Advances in database technology
- Qureshi, Khubaib Ahmed (2005), Enterprises Application Integration, IEEE --- 2005 International Conference on Emerging Technologies, Septembiiber 17-18, Islamiabad
- Rahm, Erhard; Bernstein, Philip A. (2001) A survey of approaches to automatic schema matching, December 2001 The VLDB Journal — The International Journal on Very Large Data Bases, Volume 10 Issue 4
- Reiersgaard, Nina; Salvesen, Hilde; Nordheim, Stig; Päivärinta, Tero (2005), EAI Implementation Project and Shakedown: An Exploratory Case Study, Proceedings of the 38th Hawaii International Conference on System Sciences – 2005
- Rockart, JF (1979) Chief executives define their own data needs. Harvard Business Review 52(2), 81–93.
- Salminen, Airi (2005) Building Digital Government by XML, Proceedings of the 38th Hawaii International Conference on System Sciences
- Sametinger, J. (1997) Software Engineering With Reusable Components, Springer-Verlag, Berlin Heidelberg, 1997.
- Samtani, Gunjan; Sathwani, Dimple (2001), EAI and Web Services: Easier Enterprise Application Integration?, extract from "Web Services Business Strategies and Architectures" Wrox Press (August 2002)

Reference

- Sarker, Suprateek ; Lee, Allen S. (2000), Using a case study to test the role of three key social enablers in ERP implementation, Proceedings of the twenty first international conference on Information systems, Brisbane, Queensland, Australia 2000
- Savarimuthu, Bastin Tony Roy; Purvis, Maryam; Purvis, Martin; Cranefield; Stephen (2005) Agent-based integration of Web Services with Workflow Management Systems AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems
- Schultz, David J (1979), A Case Study in System Integration Using the Build Approach, Proceedings of the 1979 ACM annual conference
- Seligman, Len; Rosenthal, Arnon (2001) XML's Impact on Databases and Data Sharing, Computer, Publication Date: Jun 2001 Volume: 34, Issue: 6
- Shanks, Graeme; Parr, Anne (2003), Positivist, Single Case Study Research in Information Systems: a Critical Analysis, Proceedings of the 11th European Conference on Information Systems, ECIS 2003, Naples, Italy 16-21 June 2003
- Shegalov, German; Gillmann, Michael; Weikum, Gerhard (2001) XML-enabled workflow management for e-services across heterogeneous platforms, August 2001 The VLDB Journal — The International Journal on Very Large Data Bases, Volume 10 Issue 1
- Shim, S.S.Y., Pendyala, V.S.; Sundaram, M., Gao, J.Z. (2000) Business-to-business e-commerce frameworks, Computer, Volume 33, Issue 10, Oct. 2000 Page(s):40 - 47
- Soejarto, Alex ; Von Uechtritz, Michael; Pring, Ben (2009), Magic Quadrant for ERP Service Providers, North America, 4 February 2009 , Gartner RAS Core Research Note G00163112
- Standish Group (2004), CHAOS Report, Retrieved May 12, 2009, from http://standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf.
- Stonebraker, Michael (2002), Too Much Middleware, March 2002, SIGMOD Record , Volume 31 Issue 1
- Stumptner, Markus; Schrefl, Michael; Grossmann, Georg (2004) On the Road to Behavior-Based Integration, 2004, First Asia-Pacific Conference on Conceptual Modelling, (APCCM2004), Dunedin, New Zealand. Conferences in Research and Practice in Information Technology, Vol. 31.
- Sumner, Mary (1999), Critical Success Factors in Enterprise Wide Information Management Systems Projects, SIGCPR '99 New Orleans LA USA
- Sward, Ricky E.; Whitacre, Kelly J. (2008), A Multi-Language Service-Oriented Architecture Using an Enterprise Service Bus, ACM SIGAda Ada Letters, v.28 n.3, December 2008
- Tellis, Winston (1997), Introduction to Case Study, The Qualitative Report, Volume 3, Number 2, July, 1997
- Themistocleous, Marinos; Irani, Zahir ; O'Keefe, Robert M. ; Paul, Ray (2001), ERP Problems and Application Integration Issues: An Empirical Survey, Proceedings of the 34th Hawaii International Conference on System Sciences - 2001
- Tolk, Andreas; Turnitsa, Charles D. (2007) Conceptual Modeling Of Information Exchange Requirements Based On Ontological Means, Proceedings of the 2007 Winter Simulation Conference
- Trotta, Gian (2003), Dancing Around EAI 'Bear Traps', ebizQ , 12/15/2003
- Walsham, G. (1993) Interpreting Information Systems in Organizations, Wiley, Chichester
- Wendt, Thomas; Birgit, Birgit; Winter, Alfred (2005), Assessing the integration of information system components, November 2005, Proceedings of the first international workshop on Interoperability of heterogeneous information systems
- Wyss, Catharine M.; Robertson, Edward L. (2005) Relational languages for metadata integration, June 2005, ACM Transactions on Database Systems (TODS), Volume 30 Issue 2
- Wyss, Catharine; Van Gucht, Dirk (2001) A relational algebra for data/metadata integration in a federated database system, October 2001, CIKM '01: Proceedings of the tenth international conference on Information and knowledge management
- Yin, Robert K. (2002) Case Study Research, Design and Methods, 3rd ed. Newbury Park, Sage Publications
- Yu, Jin; Benatallah, Boualem; Saint-Paul , Regis; Casati, Fabio; Daniel, Florian; Matera, Maristella (2007), A Framework for Rapid Integration of Presentation
- Zaitun, A. B.; Yaacob, Mashkuri (2000), Success Factors to System Integration Implementation: More Technically Oriented than Human related, ICMIT 2000
- Zellou, A.; Chiadmi, D.; Front, A.; Giraudin, J.P (2004) The E-government Platform Based On A Reused Components & Mediation, 2004. Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications, Volume , Issue , 19-23 April 2004 Page(s): 23 - 24
- Ziegler, Patrick; Dittrich, Klaus R. (2004) Three Decades of Data Integration – All Problems Solved? IFIP 18th World Computer Congress, Topical Sessions, 22-27 August 2004, Toulouse, France

Appendix A: Case 1 Data Collection & Preliminary Analysis

A.1 Data Sources

Documentations

A large number of documents for the GIS integration projects have been reviewed. They are dated from the early stage in 2003 to the most current date, as some maintenance work is still being carried out. The documents are ranged from business case, presentations, and requirement specification to system technical design. Since most of the people creating the documents have left the department, these documents are important source for the case study.

Some of the documents reviewed for the case study are listed below:

- Instructions for creating GIS map (by GIS analyst)
- Functional Spec - Linking GIS and SAP together (by system analyst A)
- AMIS-GIS Integration (by system analyst B)
- GIS Integration Issue and Enhancement (by GIS integration project manager)
- Permissions GIS Integration Software Specification.doc (by system analyst B)
- GIS Business case (by GIS manager)
- GIS IMU Presentation (by GIS developer A)
- GIS notes for technical architecture group.doc (record of group discussion)
- GIS Integration Team Discussion (record of group discussion)
- Permissions GIS Integration (by system analyst B)
- Permissions GIS Slideshow (by the GIS analyst)
- GIS Integration Notes (by developer B)
- BioWeb Spatial Integration - Business Case (by GIS manager)
- Term of reference - BioWeb Spatial Integration (by GIS manager)
- GIS - Business Proposal (by CIO and GIS manager)
- ArcGIS Server versus eView (by GIS analyst)
- Spatial Integration Software Solution (by GIS developer A)
- Permissions Spatial Integration Application Proposed Solutions (by GIS analyst)
- Permissions Spatial Integration - Business Specification Iteration 0.6 (by system analyst B)
- Permissions Spatial Integration Software Specification (by developer B)
- GIS Integration Presentation (by developer B)

Permissions - Workshop GIS Integration Tool (by GIS analyst)
GIS Integration Proof of Concept Prototype Review (by GIS analyst and business representative)
Permissions GIS Integration Software Design (by developer B)
ArcIMS documentation (from the GIS vendor)

Source Code

There are a number of standalone systems developed for the GIS integration, plus many modification works done to other existing business applications. A lot of source code has been looked into and they have provided lots insight information for system design and other technical details. Source code is for the following areas:

- SQL databases (by GIS database administrator, GIS developers B)
- GIS server (including ArcIMS map services) (by external contractors and the GIS developer B)
- Dynamic HTML Application (DOCgis) (by GIS developer A and GIS developer B)
- ASP.net Applications (including GIS Viewer, Permissions and Land Register) (by developer A and developer B)
- DocGDI batch job (by GIS developer B)

For example, in the SQL databases, there are many objects and items created for the integration in multiple servers and databases. It would be unnecessary and impossible to include all the sources within this document. The followings are some database object in NGDB1 (one for the core GIS databases) for the integrations:

TABLES (with duplicated data from other business systems)

- RPT_BWHerpetofauna
- RPT_BWObservation
- RPT_LRConservation

VIEWS (built on linked tables from other business systems)

- MG_PER_PERMISSIONS
- MG_TER_CONSUNIT
- MG_VM_PER_Permissions

- MG_VM_TER_CONSUNIT

STORED PROCEDURES (for importing data from business systems)

- IMS_BioWebHerpetofaunaReporting_Update
- IMS_BioWebObservationReporting_Update
- IMS_LRConservationReporting_Update

Interview

One interview has been conducted with the original business sponsor for the Permissions system. He was the main driving force for the new GIS Viewer integration project. The summary of interview record was attached at the end (section A.4).

System Testing

The researcher has done some testing on the GIS integration systems that are available on the organization's intranet. The two different integration systems were compared and screen shots were taken and included in the following sections.

A.2 Summary of Findings

Project Summary

The GIS integration had been a long running and multi-phases project(s). The research would not have an in-depth coverage of the earliest stage when only ESRI software was used. The approach had led to the development DOCgis and would be covered by the DOCgis stream in the case. There are two main streams of GIS integration applications running concurrently in the agency. They are DOCgis and GIS Viewer. They are both web-based applications running on Microsoft IIS server and IE browser.

First GIS Integration: DOCgis

Overview

It is one of the original and still important applications for users to retrieve integrated GIS information. It has one main user interface as the following screen shot. It allows business data to be added as extra layers, such as Biodiversity as shown on the screen. Business data and all other GIS data can then be queried or shown in the map within the interface.

System Components

The DOCgis is a map viewer browser application comes with the ArcIMS software package from ESRI. The GIS team has customized the browser interface to make its 'look and feel' like one of the agency's application. The following home page screen shot has been in such format since it was released in 2003.

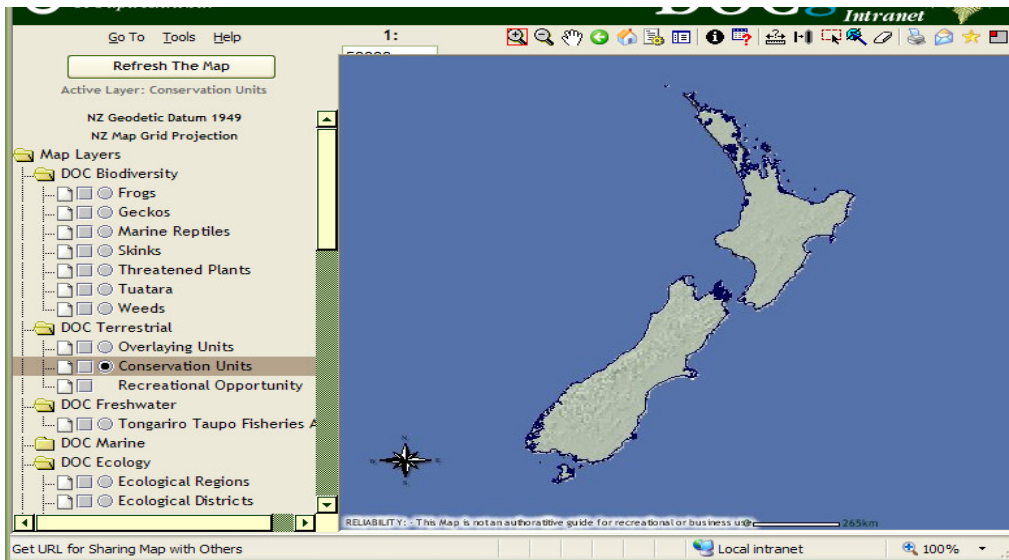


Figure A. 1 DOCgis Main Screen

In the earliest phase before the DOCgis interface was released, the integration of GIS was established by importing business data into the central GIS database (NGDBs) by using data import stored procedures. Users had to install desktop client application such as ArcCatalog and ArcReader from ESRI on their individual PC to work with the GIS and business data sitting on the GIS database server. Such approaches are still in use, especially for technical staff in the conservancy offices outside the head office, as they feel they can get more accurate and up-to-date data in that way and the tools have powerful features they need. The problem is all those tools require expensive license fees. The department can only afford a small number of such licenses, and is currently in process of reviewing the license agreement and their purchase.

The DOCgis was the first web-based GIS application available to all staff over the internal network. It was triumphed as one of the major achievement at the time when it was release in 2003. Its main features include displaying multiple business data (layers) in a map and allowing users to do some queries on those business data.

The following is a screen shot showing a map with layers of overlaying units, conservation units and recreational opportunity, which are selected by the user.

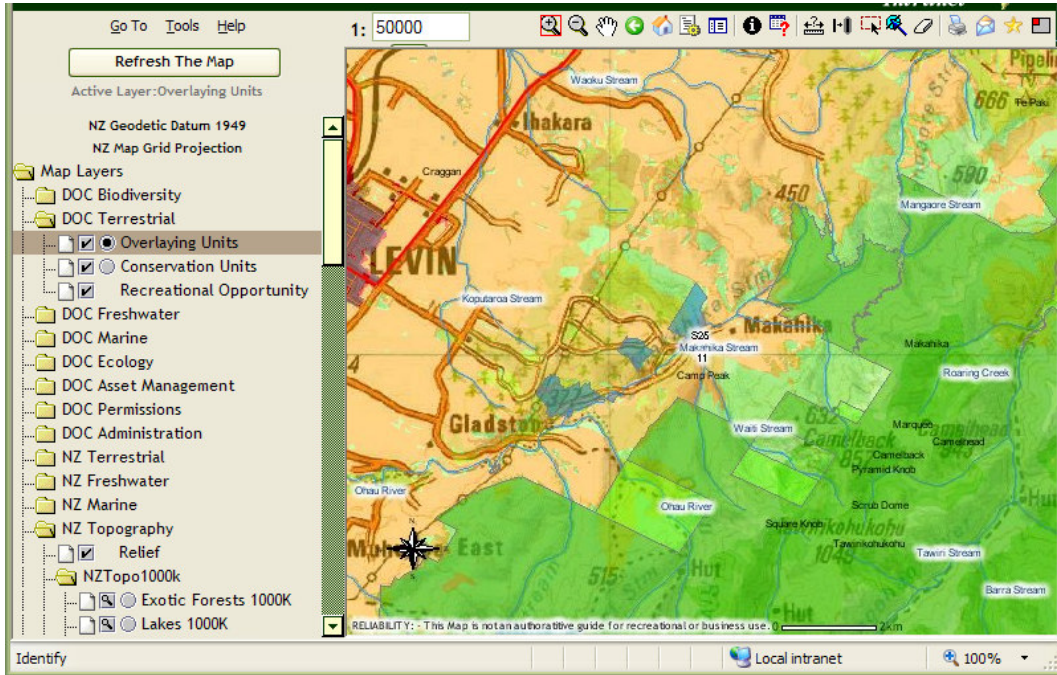


Figure A. 2 DOCgis Map Screen

Spatial query window from DOCgis:

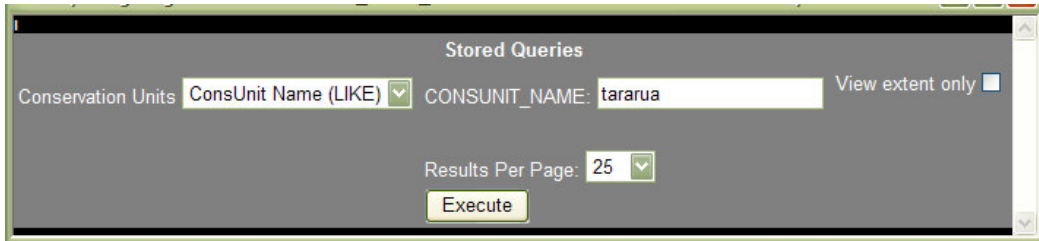


Figure A. 3 DOCgis Spatial Query

The following screen shot contains a simple table displaying conservation unit data which is requested by the user.

The screenshot shows a web browser window titled 'Query/Selection Results - Windows Internet Explorer'. It displays a table of Conservation Units with the following data:

Rec	CONSUNIT_NO	SOURCE	CONSUNIT_NAME	LEG_CODE	MODIFIEDDATE	LEG_DESC	LEG_GROUP_CODE
1	S28001	Wellington	Tararua Forest Park	CACP	1214562242000	Conservation Park - s.19 Conservation Act 1987	D
2	T24584	Wellington	Tararua Gravel Reserve	RALP	1223308730000	Local Purpose Reserve - s.23 Reserves Act 1977	D
3	F470029	Southland	Tararua Cemetery	RALP	1184689291000	Local Purpose Reserve - s.23 Reserves Act 1977	V

Below the table, there is a link: [Zoom to these records](#)

Figure A. 4 DOCgis Query Result Sample

Behind the scenes, DOCgis uses two methods to integrate business data into the GIS database with all the geometry and image data. First, as it has been using from the very beginning, it imports data from various business databases. There were limited sets of business data acquired in this way and they are disconnected from their business repository. All standard layers data such as NZ terrestrial and place name are retrieved in this way. The main issue of such approach were that the GIS database was getting bigger and bigger as duplicated data have been created. The data importing process also need to be carefully looked after and they had and can create major overload on the server hosting and running the process. Second, in later stages, business data were linked to the central GIS database by using views based on linked tables. DOCgis then works on these data, combined with geometry data in GIS database and other business data through views created in the GIS database. More and more business data could then be integrated to the data sets using this method. The main issue of the second approach is that the GIS database is strongly tied with the business database through linked tables and views. Such linkages are established in very lower level of the database objects and they are hard to discover and maintain by others except the original creator. There have been problems when changes in the business databases had caused errors in the GIS database and DOCgis. When it happens, any operation on the affected business layer by the DOCgis would generate run-time system errors. A recent incidence report showed that such problem was only detected in production, as the development team of the business system wasn't aware of any table level linkage to the GIS databases when they move the system database to a new database server. An incidence report was created because of the high severity of the problem, but there is no guarantee it will not happen again as those low level linkages are hard to detect and maintain.

National Enterprise GIS System Architecture:

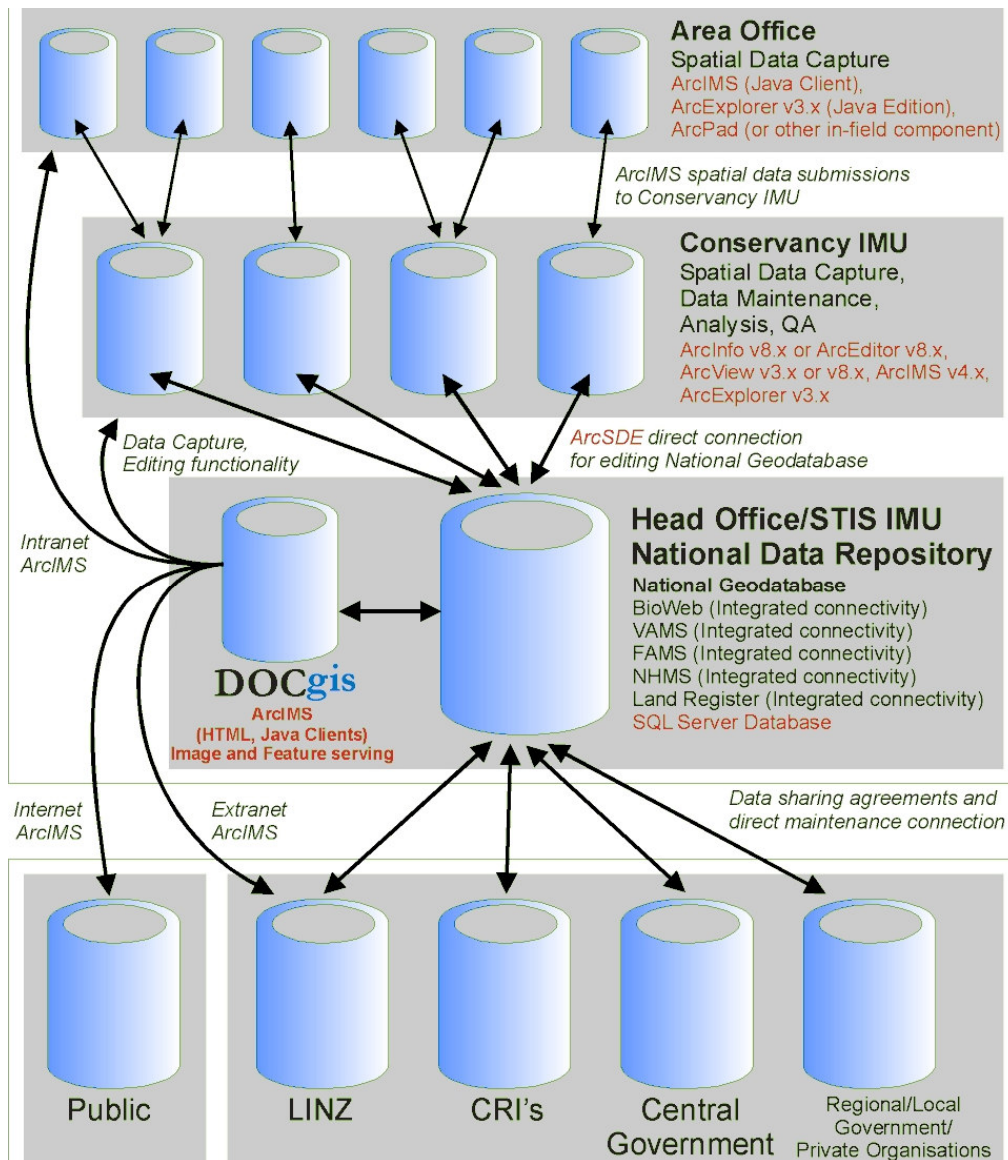


Figure A. 5 DOCgis Centred National Enterprise GIS Architecture
 (Extracted from GIS Integration Team Discussion Friday 14th 2003.ppt)

As quoted from the document “GIS Integration Team Discussion”, “the platform is called BIP ‘BioDiversity Information Platform’, it is an ArcIMS based Spatial Viewer as an off shelf product. It works using data stored in and served directly from the National Geo-spatial Databases (NGDBs).”

In summary, as the central part of the national enterprise architecture, DOCgis is a customized, HTML based web browser application from ESRI and work solely on the GIS databases. The integration is purely data integration and is done solely on the

database level, either by importing data from other business systems, or linking tables from the other databases.

Currently Status

DOCgis is still one of the core applications maintained by the GIS team, although no major change has been done on the application since 2003, as it has been an out-of-date off shelf production. Any small change may incur large amount of work and the current GIS team does not have required skills to do so anyway after the departure of the original developer. Currently, it includes a few business layers with data from their business systems, including the following list:

- Permissions
- Threatened Plants
- Weed
- Frogs
- Conservation Units

However, new data importing applications such as DOCgdi has been created since 2008 as it is indicated in the source code. It was found by the DBA on the production database server that the DOCgdi was deployed as a scheduled task running every five minutes for querying and importing data to the GIS databases. Although no major change could be identified for DOCgis from the source code, there are work done on the database as there are database change release records found in the production release history folder. It seemed the GIS team has been doing some maintenance work on the database side to keep the DOCgis updated with its data source.

Second GIS Integration: GIS Viewer

Overview

In 2005, GIS integration requirements for the Permissions application kicked start a new phase of GIS integration development for the department. Because of the lack of features and flexibility of the ‘old’ DOCgis application, new options and technologies had been evaluated. After evaluation of a series of products including eView, ArcEditor, and ArcGIS server, it was concluded in the end that the department would develop the new integration structure including the new user interface developed in

house. It was a decision after a long period of extensive research and consultations for over a year.

System Components

The core component of the new GIS integration is the new GIS Viewer. The interface design of GIS Viewer is similar to the existing DOCgis with all the existing functions, but is more users friendly and based on user feedback. Most importantly, it provides many new functions such as spatial queries using free-drawing user defined area or existing defined area from business layers. The spatial query was one of the crucial business requirements. Behind the viewer interface, while it still uses the GIS databases for standard GIS data such as place name and topo layers through the ArcIMS map service, the new GIS Viewer retrieves other business data with XML web service provided by respective business systems. The ArcIMS map service is mainly responsible for providing map images generated from the core GIS database; other business data are retrieved through dedicated XML web services. Another new feature of the GIS Viewer is it provides hyper-links for functional integration between the spatial viewer and business applications. The links are enabled through special XML web service. In addition, the GIS Viewer works on an XML configuration file; it is possible to created dynamic links to various parts of other business system in the GIS Viewer content. So far the integration using GIS Viewer has encompassed Permissions and Lang Register.

Main GIS Viewer screen

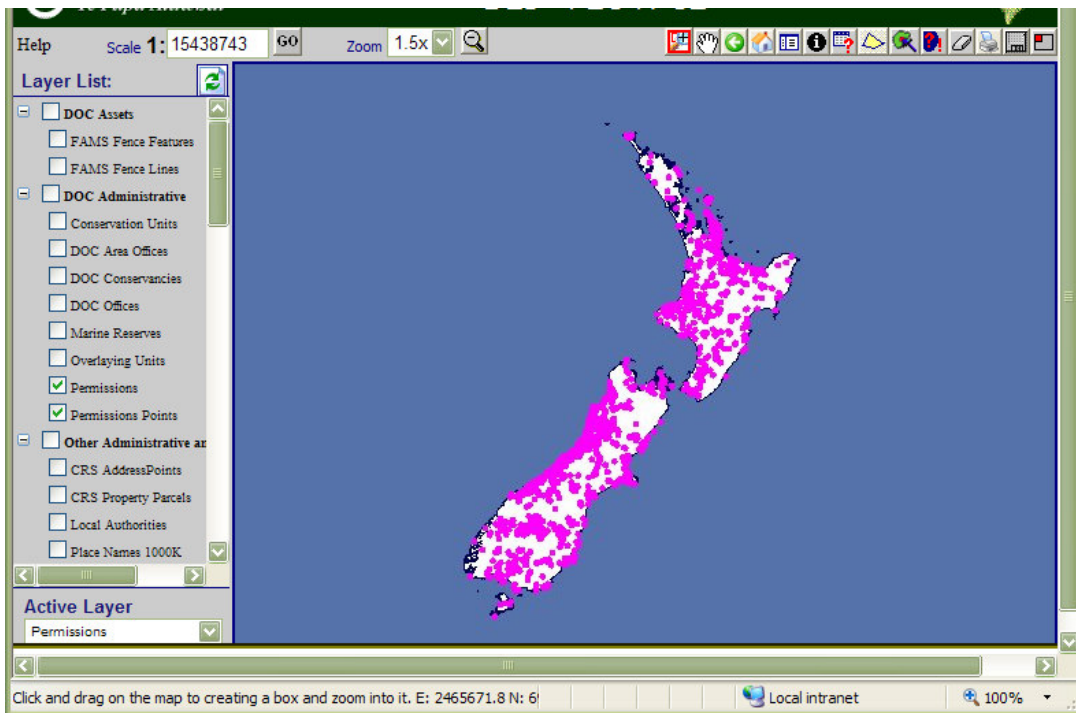


Figure A. 6 GIS Viewer Main Screen

Within the main screen of the GIS Viewer:

- 1) The main part of the screen is for displaying the map consisting of multiple layers of information.
- 2) The layers would be selected from the lay list on the left.
- 3) There is a toolbar on top of the map for users to initiate operations on the map, including zooming, panning, and spatial querying.
- 4) The bottom part of the screen is for displaying spatial query report, which is none at the captured screen.

Detail of all the features is out of scope of this research, but the import ones are explained briefly in the following sections.

Spatial query window using free-drawing user defined area (highlighted in yellow):

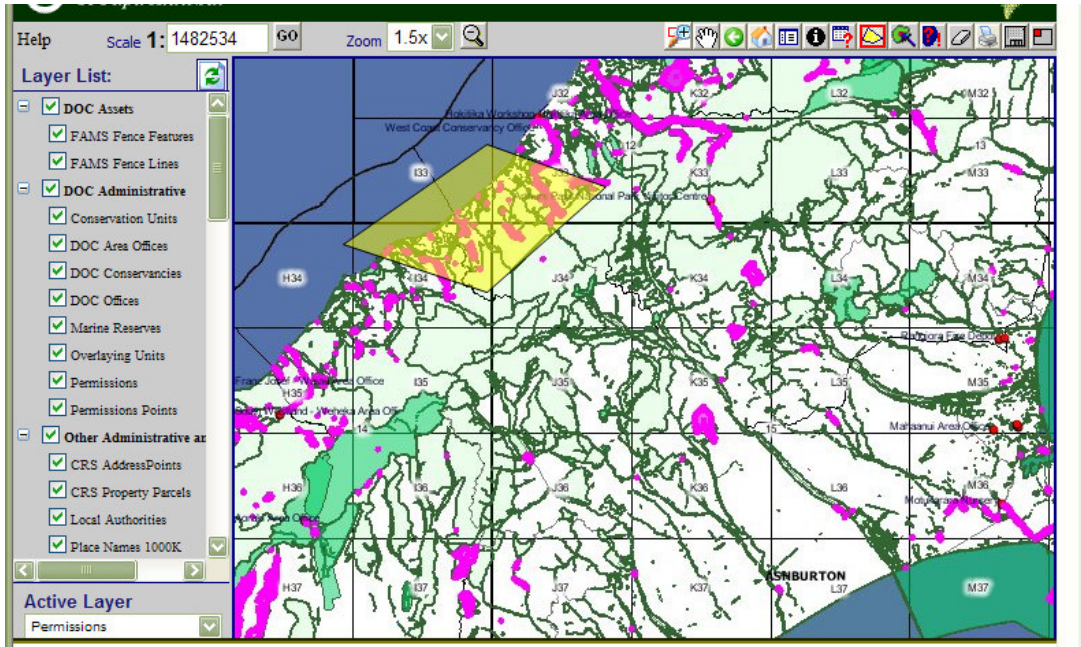


Figure A. 7 GIS Viewer Spatial Query on User Defined Area

Spatial query window using existing area (highlighted in yellow) from business layer:

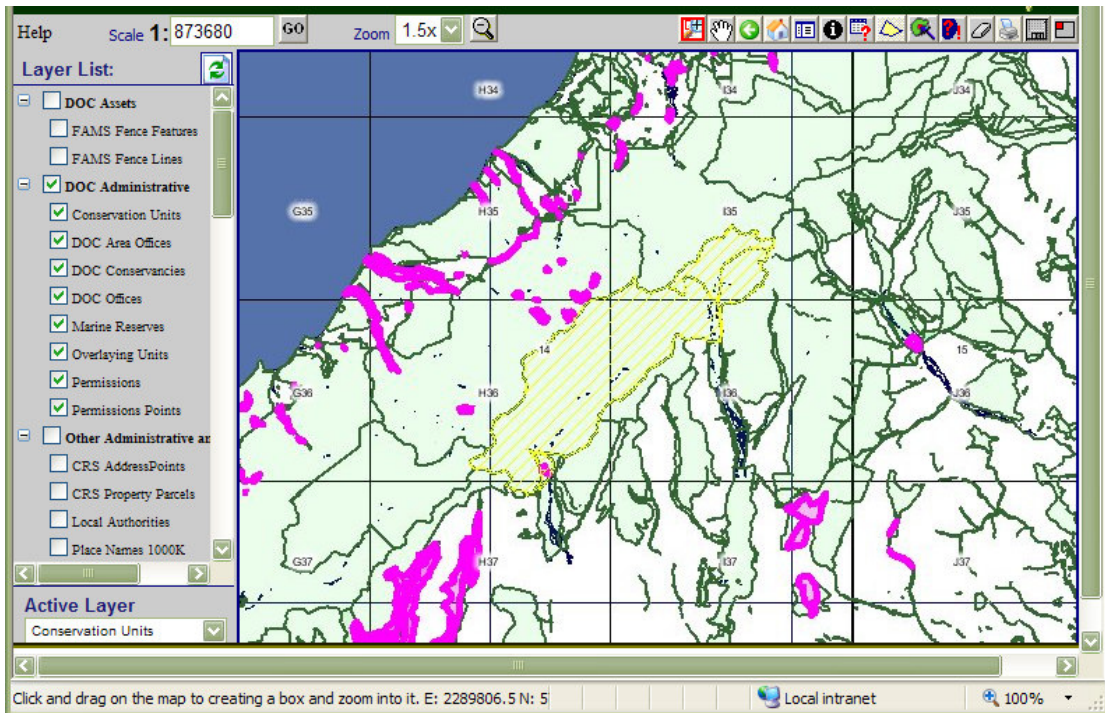


Figure A. 8 GIS Viewer Spatial Query on Existing Defined Area

Spatial query report sample screen (The highlighted text is hyper link to the Permissions business system):

(The style and content of the report are defined in the XML configuration file; all can be adjusted without any change to other part of the system.)

Permissions Location (3)		
Location	Permission	Status
Albertburn Conservation Area	Fishing numerous rivers in wider Lake Wanaka/Hawea area	Active
Albertburn Conservation Area	Heliskiing - Albertburn block ex tender	Active
Albertburn Conservation Area	Part Section 3331 Adj The Albert Burn Block XI Wilkin SD area 90 ha	Active

DOC Conservancies (1)			
CREATEDDATE	MODIFIEDDATE	GIS_USER	CONSERVANCY
		sdeadmin	Otago

DOC Area Offices (5) ...
Local Authorities (3) ...
Ecological Districts (10) ...

Figure A. 9 GIS Viewer Spatial Query Report Result Sample

Business data captured in the GIS viewer (the screen is linked directly from Permissions):

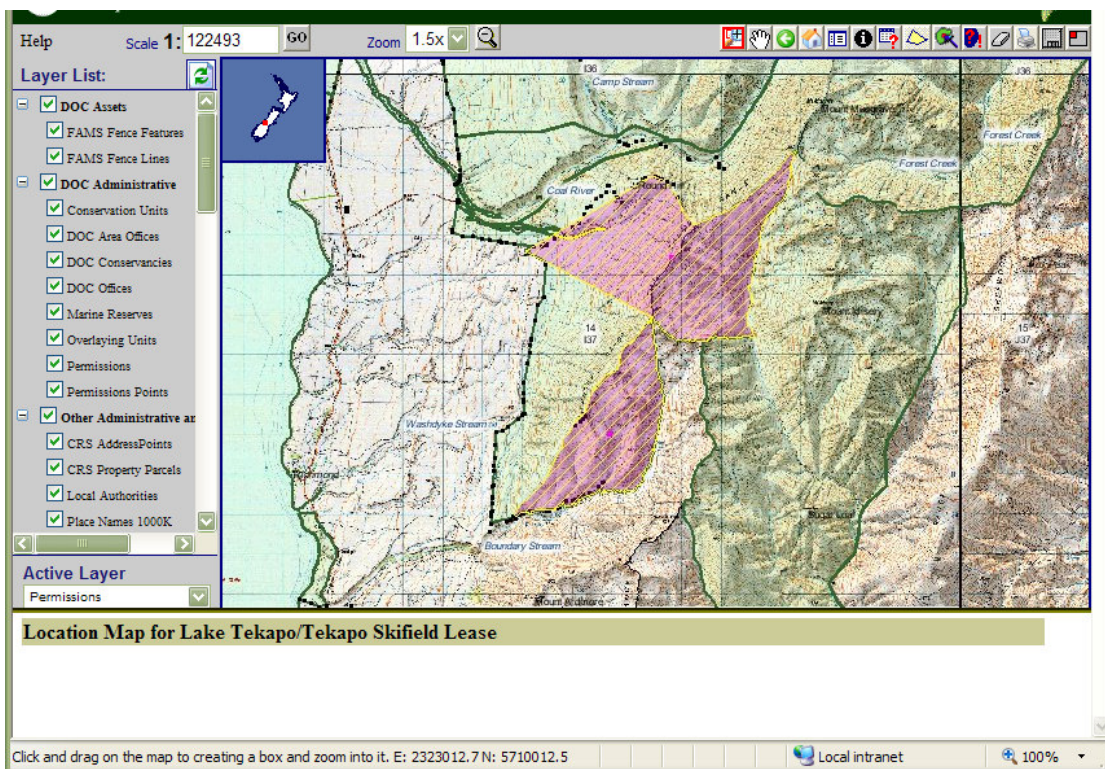


Figure A. 10 GIS Viewer Map Sample

Integration functions have also been developed and included in the Permissions business system. Most of them are displayed as hyper links behaving in the same way as those in the GIS Viewer. Some screens of the business system have been modified to accommodate the hyper links and to display special GIS integration information.

Integration A

In the following screen, the ‘GIS Integration’ column displays the map status of the permission record, they can be ‘Fully’ which means all locations for the permission are captured and would be displayed, they can be ‘Partially’, which means only some locations are captured. If no location has map data, the cell will be empty. By clicking on the cell which is a hyperlink, the user opens up a new GIS Viewer window such as the picture above.

Holder Legal Name	Permission No	Status	GIS Integration
A D	LGA 0273	Active	Fully
A D	LGA 0227	Active	Fully
Ltd	LMS 250	Active	Fully
B	11/LGA0189	Active	Fully
New Zealand Limited	WC-14964-RES	Expired	Partially
New Zealand Limited	WC-15216-RES	Expired	Partially
	LSM 0277	Expired	Fully
	LGA 0220	Active	Partially
	S144	Active	Fully
	11/L0673	Terminated	Fully

Figure A. 11 Permissions/GIS Integration Screen Sample 1

Integration B

In the following screen, GIS integration information is displayed for a particular permission record in individual activity and location level. Hyperlinks are enabled in respective level, such as individual location, within conservancy boundary and the entire permissions.

PAL	Activity	Location	All year
Auckland (3 locations) ...			View In GIS
Canterbury			View In GIS
14996 Short Walks		Hooker Valley Track	GIS <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Nelson-Marlborough			View In GIS
17675 Day Walks		Queen Charlotte Track	GIS <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
14994 Short Walks		Abel Tasman Coastal Track-Great Walk	GIS <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
14997 Short Walks		Kaikoura Peninsula (Kean Point) Scenic Reserve	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
15001 Short Walks		Nelson Lakes National Park, St Arnaud Mgmt Area	GIS <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Southland (4 locations) ...			View In GIS
Tongariro-Taupo			
15000 Short Walks		Short Walks- Tongariro National Park	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
West Coast			View In GIS
15004 Short Walks		Alex Knob Track	GIS <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

Select activities and locations to add

Guiding | Abseiling/Rap Jumping

Please select an Area Office...

New Location | Show Levels | Clear Choices | Add

Figure A. 12 Permissions/GIS Integration Screen Sample 2

In 2007, another major business application Land Register was integrated with GIS Viewer in the similar fashion as Permissions.

GIS Integration System Architecture (Re-created from the source code)

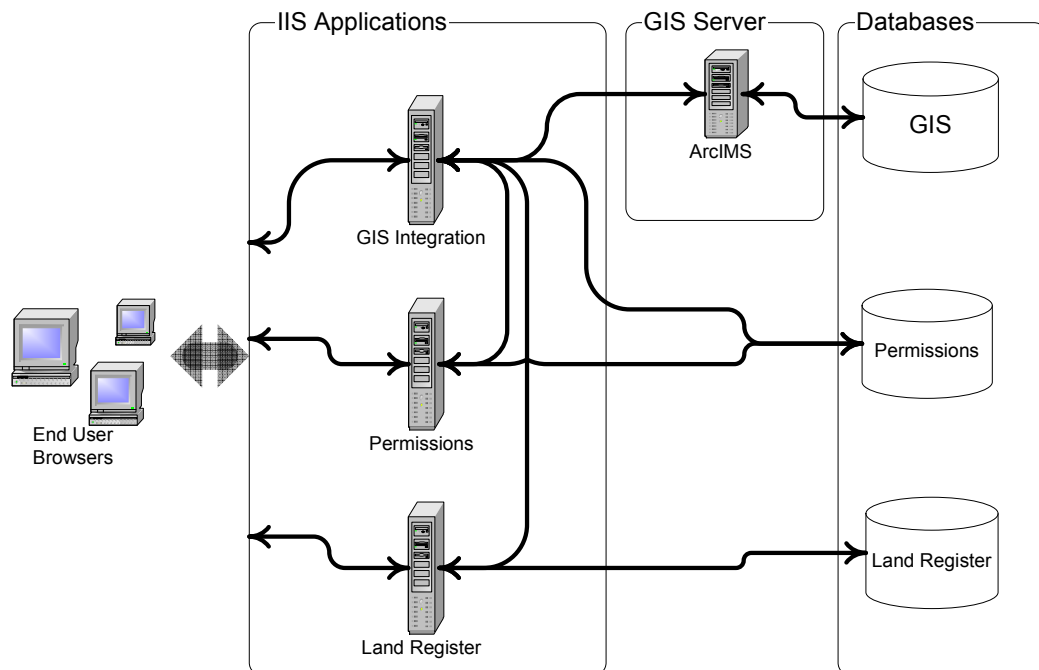


Figure A. 13 GIS Viewer System Architecture

In summary, the GIS Integration (GIS Viewer) system is not just an in-house re-development of the first integration (DOCgis) with extra new functions. Firstly, in data integration, the new GIS Viewer uses dedicated XML web service to retrieve data from individual business system. It has not only reduced the coupling and dependency in the database level, it has provided flexibility to the business system and improved the maintainability of the integration links. Secondly, the new GIS Viewer has enabled dynamic hyper links with business systems. It has provided some powerful functional integration functions to allow users to explore various business systems in an easy way by following the links. The extensive usage of XML web service and XML configuration has streamlined the integration development and deployment; it has created a rapid development model ready for more future integration.

A.3 Preliminary Analysis

Apart from the general data analysis method of this research thesis as they are mentioned in the previous section, there are some particular aspects for the GIS integration projects as followings:

- 1) There were a large amount of source code for both DOCgis and GIS Viewer, the research had to go through them and re-created the architecture diagram.
- 2) The source code was also used to establish chain of evidence in many areas, such as the integration design and their advantages and disadvantages.
- 3) Comparisons have been made in many areas between the two integration approaches.

Case Study Questions:

- 1) *What is the GIS integration project from different prospective within the department?*

From the business point of view:

As stated in the GIS Business Case, “This system provides department wide access to spatial data and GIS tools, which are essential to both Integrated Conservation Asset Management and Measuring Conservation Achievement (MCA) programmes (Conservation 2001). The Integrated Conservation Asset Management programme supports the integrated management of all classes of conservation assets and consists of the following systems and databases:

Bioweb – Natural Heritage Assets Management System (NHMS)
Visitor Assets Management System (VAMS)
Fencing Assets Management System (FAMS)
Historic Heritage Assets Management System (HHAMS)”

GIS integration is essential to maximize the benefit the department can get from all the existing software investments the department has made, as it would enable ordinary users of any business systems to access GIS (map) data as long as the business system is included in the integration. This view was reflected in many other document including the business requirements and system design.

According to the Permission GIS integration requirements document, it is essential that users of Permissions system can associate their business data with location, area and other GIS information. For example, a guiding permit must be associated

with area or location where it can be operated. GIS integration with online data integration and functional integration is the most effective way to allow users to achieve that. Without the integration application, users of business system such as Permissions have to use multiple software tools or rely on other specialist to get any map or spatial report. It was also recorded during the GIS Viewer prototype workshops that, attending users were impressed with the integration functions the system could provide and they looked forward to its implementation.

From the system development point of view:

The GIS team has tried to build a national enterprise GIS system since 2002. As shown in its architecture diagram and other documents, they have been focusing on the backend databases and relying on lots off-shelf product including ArcReader and ArcEditor. The GIS application (DOCgis) have been developed and included in the enterprise architecture as one of the key user interface of the integration components, which was also shown in its architecture diagram. In many other documents including the GIS team discussion document and presentation, DOCgis was promoted as the central piece of software to access all GIS (map) information and associated business data.

From the organization structure and individual job descriptions, the IT development team's job is always to develop processes and software systems to help the business, such as enable the business to achieve their goals and carry out their tasks in an effective way. The GIS integration system should include effective data integration as well as functional integration as stated in the solution design document. The new GIS Viewer and all the components and source code have been created to fulfil such jobs.

2) *What are the goals for the project?*

From business point of view

They were clearly defined in a number of document including the business case and proof of concept review for the new GIS integration (GIS Viewer). They were:

- i. To allow GIS data (maps) to associate with business data, and vice versa.

- ii. To allow business users to access GIS data such as maps directly from their business system. Without the integration, business users have to ask the specialist GIS users to produce the required maps.
- iii. To allow users to run spatial queries and reports based on maps and locations. Without the integration, only specialist GIS users can do such queries with specialist GIS software.

For the old DOCgis integration, the business goals were also mentioned in many documents. They tended to be of high level and not specific. In one of the GIS presentation document, they were defined as “Using geo-referenced data a GIS can provide answers to questions about a specific location or the distribution of a phenomena, the changes that have occurred since a previous viewing, the impact of a specific event, or the relationships and systematic patterns of a region.” It was also said the integration of specialist business databases would assist with decision-making and policy analysis in a discussion document.

From system development point of view

- i. As from GIS Viewer system design document, the project was to use the latest technologies to create a generic integration solution for all business systems with the GIS software, which was shown in the architecture diagrams.
- ii. In many of the presentation and discussion documents from the GIS team, the integration was regarded as a core building block of the National Enterprise Geo-spatial Information System (GEGIS), which was clearly shown in the architecture diagram.

The goals from the GIS team were of very high level and would be ongoing for a long time before they can be achieved or completed. However the goals for the later GIS Viewer development were more basic and achievable.

3) *Why is it important to agency?*

As the agency is the government agency for administrating most of crown land, most of the business and activities of the department are associated with locations and geo-spatial information. Some of the core business systems include AMIS, Permissions and Land Register. They are all related to the land and location. AMIS

stands for Asset Management Information System, the system is for managing facilities like tracks, huts and toilet in all the national parks. Permissions system is about managing activities especially commercial business such as tourism business on the crown land. Land Register is for keeping records of those lands under the management of the department. According to a discussion with a GIS team member and information on the organization structure, GIS is an important part for the department operation, there are about 27 full time GIS specialist employed in 13 conservancies around the country. They are responsible for maintaining GIS data and providing support for other users who would require GIS information from time to time. It was stated in the Permissions GIS document, current GIS tools required extra expertise and software licenses; they were out of reach for ordinary users; the new integration of GIS with other business applications would enable all the agency users to access GIS data directly and carry out their work much more efficiently, it would potentially save a lot of money for the department from paying licensing fees for those specialist GIS end-user products. In the interview with the former business sponsor, it was argued, GIS integration was part of the department’s mandate; it would create new business opportunity. In the end, as stated in the GIS - Business Proposal document, the GIS integration solution was aimed to increase value of the department’s IT investment and reduce the total cost of ownership of GIS system including those specialist ERSI software licensing fees.

4) *What are the solutions that have been used so far?*

By reviewing all source code and design document, the architecture diagrams have been created and displayed in the previous findings section. The solutions are summarized in the following table:

Core Components	Core Technologies
ESRI SDE engine, ArcIMS map service	Commercial software
NGDBs with data integration stored procedures and views	Microsoft SQL
DOCgis user interface	HTML, JavaScript
GIS Viewer user interface with XML configuration	XML and ASP.Net, IIS server, JavaScript

XML web service interface for data and functional integration	XML, .Net Framework
Windows service for regular data update	Windows .Net Framework

5) *What are the critical success factors for the integration project?*

From reviewing project documents, especially the issues and risk register, and discussion with a system analyst, the investigator found the following factors would be the most crucial for the project:

- i. Business requirements are agreed and well documented.
 It is never easy to define and agree on business requirements for a large organization like the agency. There are many different views on a range of topics as the documents have revealed, such as the NEGIS architecture and the new Permissions GIS integration requirements. Without a well-documented requirement, any development would be very difficult if it doesn't fail in the end. As documented in a discussion, the former GIS manager said, "What is holding us up? as I understand it, they are user requirements – fragmented info/approach from different databases. We can't do just yet because we don't know what exactly is required"
- ii. Suitable software and development tools are chosen.
 The development team had been in struggle in finding suitable new development platform and tools. As it was shown in the project document, it took over a year to evaluate various options. The final decision of using AcrIMS and building the new web-based GIS Viewer was a very important milestone as it sets the direction for the rest for development. In the same discussion mentioned above, the former GIS manager also said "programmatic techniques re current apps, i.e. how to do it" was a major hurdle for GIS integration.
- iii. Projects are well managed and supported.
 Any project would not succeed without proper resource and management, particularly for projects like GIS integration that involved a number for core business systems, databases, and required specialist skills. The research has found records and documents for regular project meetings, such as the issues

and risk register document. These documents had shown it was crucial to manage the project properly and those documents were important part of the process. They helped to make sure project issues would not disappear from the 'Radar' without resolution, they were dealt with according to priority and with proper resource (responsible staff). The documents also shown there were a number of active member and contributors from the development team, including the business analyst, developer and business representative, all have created or contributed to a large number of documents and works.

iv. Business Involvement

It was an important factor for this project, as business was the primary force behind the new integration functions. The business rep had been involved from the beginning for the requirement gathering. In the prototype workshop, a group of key GIS users had participated and provided lots feedbacks, which all had been feed back to the final requirement document. The business rep was one of the main contributors to the documents.

v. User Acceptance/Training

The new GIS integration provided some complex functions. Before it was release, the business rep had created special training material and included them in the Permissions User's Guide. The guide would be accessed through a hyperlink displayed in the 'Help' screen of the GIS Viewer. For the old DOCgis system, there had been specially printed copies of the users' guide distributed to all the offices. For complex integration system like GIS integration, user acceptance and training were very important.

6) *What are issues and challenges faced by the project team?*

From reviewing the documents and discussion with various people involved, the following issues were identified:

i. Business requirements were hard to defined and agreed on.

There are a huge number of documents related to GIS from the department document storage, a search by the investigator using 'GIS' key word returned over 800 documents. There were all sorts of requirements ranged from the top level like new software licensing to the low level such as new fields in a GIS table. Many business users from many different offices across the country would have different views and requirements, but most of them may not have

much experience in GIS software. Getting clear messages to and from the business had been always a challenging task and needed to be handled with skills and strategies. As stated by the former GIS manager, getting clear business requirements was one of the biggest challenges. From the initial discussion document of Permissions GIS integration in 2003 to the final requirements document in 2005, it had taken over two years for the requirement to form up. The GIS integration was a long running one and over a year had been spent on trying to identify the new users' requirements. Only after the workshop and presentation of prototype, the requirements were clarified and the road to development was cleared. As argued by Richard Berntsson-Svensson, Aybüke Aurum (2006), clear business objectives, minimized scope and firm basic requirements are critical success factors for software project.

- ii. The database only approach for the integration was not suitable to complex situation.

The designs of the old GIS data integration included duplicated tables, regular data updating processes and views across databases. They were problematic. As it was illustrated in the NEGIS architecture diagram, the old GIS integration was based directly on multiple databases, which would be connected to each other in database object level, include tables, views and stored procedure processes. Further investigations into the databases by the investigator confirmed they were still in use currently with ongoing work done (i.e. DOCgdi process) to them. The design would have many problems. They created tables with duplicated data, which would get bigger and bigger. They also had created extra workload for the server to keep running those regular data updating process. It was found one of those new processes called DOCgdi was scheduled to run every five minutes on the database server. Those views across multiple databases are hard to detect and maintain, as the definitions of the views only appear as the SQL server script and they are not in any form of source control. There were many incidence reports of GIS errors caused by broken connections in databases or overload on the database server. A recently documented one was caused by a simple change of a view definition in the GIS databases.

- iii. The off-shelf DOCgis viewer is difficult to maintain and lack of some crucial features.

By looking into the source code, it could be confirmed the DOCgis was developed in basic HTML and JavaScript. It was also documented in other places including the architecture diagram. Basic HTML and JavaScript applications have become out of date and replaced by other advanced technologies included ASP .Net since 2002. The DOCgis application became even harder to maintain or enhance after the original GIS developer left the department in 2004, as it was found out in discussions. The former GIS manager also conceded in many of his discussion that he was searching for new techniques for building new applications. As it was document in the Permissions GIS integration document, as the DOCgis could not provide crucial functional integration features and adequate spatial query features, the new integration needs must be fulfilled with a new GIS integration development.

- iv. A new approach needed to be chosen from many new GIS software packages and development tools.

As it was shown in many documents, many options had been considered for the new GIS integration project ranging from completely outsourcing to total rebuilt from scratch. The development team had called in some experts for consultation and also carried out its own investigation. Evaluations have been done on many tools and software packages including eView, Spatial Media and ArcGIS server. The process had been stretched over a year as the relevant documents had shown. The development had difficulties from many factors including defining scope of project and getting to know the new tool and software. Many documents were found especially created for evaluating and comparing those different options.

- v. The old GIS team was mostly isolated from the rest of development team. The DOCgis was customised from an off shelf product of ESRI, the structure and files of source code was found matching those from the original map viewer from the vendor. By checking on the changes of files and other documents including the DOCgis presentation, it was found that the customisation work was done solely by the GIS developer of the GIS team at the time. The GIS developer has left the agency and moved away from

software development since 2004. By checking the source code, no QA or source control mechanism was found to have been used. Until recently, no general software development and deployment procedure was followed by the GIS team. While the rest of the development team have followed formal procedures since 2002. From discussions, the investigator heard that the former GIS manager had argued in many meetings that GIS was specialist software and the general development rules for the rest of IT department did not suit it. No evidence was found why the development of DOCgis was different from other web-based developments in the agency, as they all used similar techniques such as html and JavaScript. However, it was found the GIS team only had done small development work and worked most with existing software package from ESRI, and it had been trying to keep it that way. Although it was concluded before development of the new GIS integration, the DOCgis would not be an adequate solution for GIS integration with business system like Permissions as stated in the business case document, the former manager of the GIS team had never agree to stop using DOCgis and to fully support the new GIS Viewer. As the former business sponsor said, there were some politics and personality issues with the project team.

7) *What are the major technical issues and solutions for the integration project?*

From reviewing the documents and discussion with various people involved, the followings are the major challenges and their solutions during the GIS Viewer development:

- i. Business requirements were hard to defined and agreed on.

While there was no document found for the old DOCgis on how business requirements were collected, from a chain of documents including the initial design, workshop report, and requirement document, it was found the new GIS integration development team had created a functional prototype user interface and presented it in a workshop attended by potential users. The workshop was organized on the early stage of the development and was well documented. User feedbacks are then used to formalize the requirement document afterward. Prototyping is one of the very effective development methods. As argued by John Crinnion (1992), prototyping is the best known

and almost certainly the most important of IS development techniques. One of the original definitions of prototyping was to build a working model of part of the proposed system to help us to understand better the real requirement.

Since then, the method of prototyping has evolved to have different forms and paths. Some prototypes are of throw-away type and are built purely to identify basic requirements and illustrate to users. They are built in ‘quick and dirty’ manners and discarded afterward. Other prototypes are evolutionary and built in more structured manners; they become the base for the development for the real systems. John Crinnion argued that the later approaches were becoming more and more favoured by the industry. The prototype of the GIS Viewer has evolved into full development and eventually into production. The integration requirements were clearly documented after the prototype demo and user workshop.

- ii. New development platform and tools needed to be chosen from many GIS software packages and technologies.

It was found for the new GIS Viewer development, after over a year of evaluation, many consultations and presentations from experts and potential vendors, the development team had then done a proof of concept system using different approaches including off shelf software packages, third party components and a pure .Net in-house development. With the successful results from the proof of concept system, the development team then chose to build a full system in-house with the .Net framework. In a document of proof of concept, two different options was outlined and compared, one used a third party component, the other use existing .Net library to build the entire system. The third party component seemed to work initially, but then bugs were encountered, the communication with the vendor and getting the bugs fixed was time consuming. The document recommended the in-house development version, as all features of the third party component had been developed in the in-house version. In short, proof of concept is a quick realization of a certain method or idea(s) to demonstrate its feasibility for full use. It is regarded as a milestone on the way to a fully functional prototype. Like prototyping, it is one of the most effective methods in system development. As argued by Odysseas Pentakalos (2008), proof of concept should be considered before

one settled on the design for the final deliverables. It provides many benefits including understanding of the capabilities and limitations of new technologies. In this case, the method was used to evaluate potential techniques and third party components. The third party component was found cumbersome to use, while the in-house .Net component was found quickly built, easy to maintain and enhance, hence more suitable for the new integration system.

- iii. New approaches need to be identified for data and functional integration across multiple systems.

For the GIS Viewer development, the team had to find new ways to improve the data integration and implement new functional integration. The data integration in DOSgis was achieved in low level database objects, since all other business data are from within the department and in the same SQL server database format, such approach could be easy to set up and could be effective. As argued by Erik G. Nilsson Else K. Nordhagen Gro Oftedal (1990), all business system within the same organization can be viewed as the "total" data basis and functionality of the organization. Even so, the strong coupling effect of such approach has created many issues, as the sub systems are from different systems and are maintained by different people. During the literature review, a plenty of theories and frameworks have been discovered and argued for better ways for integration. As argued by Suresh Damodaran (2004), XML is ideal for transferring data between different system, because of the text base and self-describing characteristics of XML. A XML web service based data integration framework would greatly reduce the dependency in the database level and improve the maintainability and reliability of the integration. As argued by Airi Salminen (2005), data integration is a portal to provide directions and linkages to various systems. It should not be strongly tied to the sub-systems or even substituted them by making duplicated copies. For the new GIS Viewer, XML web service had been extensively used for data integration, no new database object level connection had been found created.

The new GIS viewer also includes functional integration with business systems. Many papers had argued for the advantages of using XML for

functional integration. As argued by Brahim Medjahed, Boualem Benatallah, Athman Bouguettaya, Anne H. H. Ngu, Ahmed K. Elmagarmid (2003), a large number of contemporary B2B interaction standards are based on XML. XML has become the default standard format for data and functional integration because of text base and self-describing characteristics. Again, a XML web service based approach would provide the best flexibility for the functional integration while put little constrains on the sub-systems. It was found the new GIS Viewer used XML web service as portal to interact with other business system, so were the business systems including Permissions and Land Register.

8) *How has XML been involved in past and current system integrations?*

By going through the development projects and source code, XML was not found in the old DOCgis integration, but was in the new GIS Viewer system in following areas:

i. GIS Server Interface

By looking into the source code, it was found that the interface component for the GIS system (ArcIMs .Net Link library) use XML as data format for queries submitted to the server and result returned from the server. The query request XML contains the entire query statement in an XML document. The response XML contains the entire result set in an XML document too. The XML documents would be very flexible as long as they following rules set out by the ArcIMS server.

ii. Data integration

Data integration includes data traffic of dual directions between the GIS Viewer and business systems, as it was illustrated in the architecture diagram. By looking into the system design and source code, it was found some data integration settings were stored in the configuration file in XML format in the GIS Viewer system, the system could interpret them and make queries to the business system when the data is needed, such as during spatial queries and reporting. Since the settings are in pre-defined XML format, new contents for new business sub-systems are easy to be added into the integration. For getting data to the business system side, XML web service interfaces have been created for retrieving data from the GIS databases in XML format and

return them. The XML datasets are for updating the GIS status in the business systems only. All input parameters and output data are passed in XML format. They are treated as simple XML document packages and free of any system format constraints.

iii. Functional integration

As it was illustrated in the architecture diagram and the screen shot summary in the previous section, there are functional interactions between the GIS Viewer and business system in both direction. By looking into the system design and source code, it was found XML web service interfaces have been created as a portal to pass requests between the GIS Viewer and business systems. These web services take in XML document as parameter, process the data and carry out the task described in the XML document.

iv. Other Configuration

As part of standard .Net applications, there are many other settings for the GIS Viewer are stored in XML files. It was found there were a few XML files in the GIS Viewer system, including the LayTree.xml and SpatialQuery.xml. Even the recently built DOCgdi using .Net for maintaining the synchronization for the business layers in the old DOCgis was found using XML configuration files. Again, this was because there were build-in XML capacities in the .Net development tools as the developer said in the discussion, it was easy to use and content is easy to maintain.

9) *What are the reasons for not adopting XML in the first integration DOCgis?*

In the early stage of the project pre-2003, XML was not adopted. Data integration for the GIS system was done through back end database connections using linked tables, cross database views and data importing stored procedures. The exact reasoning behind such design is unclear, as at the time there was only one GIS developer who had left the department since long time ago. The approach would be the quickest and simplest way to achieve the simple data integration of the GIS system and other business systems at the time. The reasons would include:

- i. There were only a few dataset were integrated (imported) to the GIS databases at the time.

By looking into the source code and documents, it was found, in 2003 when the DOCgis was released, only a couple of dataset from Bioweb system were

included, all other dataset are static public GIS data, such as topographic and place names.

- ii. There wasn't any resource for doing more analysis or development.

According to the organization structure, there was only one developer in the GIS team. From the documentation, the GIS developer customizing the DOCgis had only been around for two years, he had not involved with any other business system. In many documents, the GIS manager had claimed his team only has "limited resources to date for a mammoth task".

- iii. The DOCgis was customized from an off shelf map viewer product from ESRI and it was not designed for using XML for integration.

By looking into the design and source code, it was found the map viewer was a simple product not designed for using combined (integrated) data source; it could only work with single backend database interface. Any changes apart from the look and feel would involve major work and that could be beyond the GIS team capability.

10) What are the reasons for adopting XML in the second integration GIS Viewer?

In 2005 when the second GIS integration was developed, the situation has changed dramatically. The GIS Viewer was a brand new development using the latest technology on the revised business requirements. Reasons for adopting XML can be summarized as followings:

- i. Since the GIS server interface component ArcIMS .Net Link library used XML as data format, the new GIS Viewer had to submit queries in XML and interpret the results in XML.
- ii. The agency and the development team were more ready than before for using XML. Since 2002 when Microsoft released its Visual Studio .Net development platform, which has heavy use of XML, the agency had used it to develop a number of business systems, including PestLink, Permissions and Land Register according to the history of source code. In the mean time, by comparing the authors of documents and source code, it was found developers for the GIS Viewer had been involved in many more XML and .Net projects than the developer for DOCgis.

Since XML and web service are part of the development tools of the Microsoft .Net platform, they had become the standard for the department's

software development as documented in the application standard authorized by the application manager. The researcher found they did provide many advantages. For example, since the data integration was centralized and defined in the XML configuration file, they were much easier to maintain and more flexible for implementation of business rules, which was an objective stated in the system design.

- iii. It was found the GIS Viewer had been developed using formal development methodologies according to a set of documents. First, the team was allowed to evaluate different product and development options. Second, formal development techniques such as prototyping, proof of concept development were adopted for identifying business requirements and choosing development tools. In the technical design document, it was concluded that using XML and web services were necessary and more effective, and the new XML techniques were tried and proven in the working prototype and proof of concept development.
- iv. The GIS viewer development team had learnt from issues in the previous integration system without XML. The new GIS Viewer integration was more flexible and robust with XML and web service approach as it reduced coupling with business systems through web service for the integration, which was stated as one of the objectives in the system design. As the old methods in DOCgis relied only on back end database connections which depends on many low level database objects such as tables and fields directly, they were of tight coupling software design which were easy to break and hard to maintain. The XML web service had improve the situation by providing a new interface and allowing the connection setting to be maintained in the XML configuration.

11) How were the decisions of the system design made?

- i. Business requirements had forced to use new design and techniques. In the Permissions GIS integration requirement document, it stated DOCgis could not be modified to accommodate the spatial query and in depth functional integration with the Permissions systems. A new GIS integration system needed to be developed using the suitable technologies, including XML.

- ii. The agency's software development standard had affected the design.
As part of the software development standard imposed after 2003 for the department, the second GIS integration project was supposed to comply with the rules that XML and web service should have been adopted where it is applicable.
- iii. The development team had followed formal development methods.
The development had chosen the design and technique through evaluation, proof of concept and prototyping.

12) How successful were XML web service integration solutions when XML was actually used?

For the business point of view, the GIS integration with XML technologies was a big success. Reasons found are as following:

- i. A lot of business users including the Permissions business manager and CIO at the time, were impressed with the powerful features of the new GIS Viewer including the spatial query and hyperlinks with business systems, as it was recorded in discussions with the developers.
- ii. Even without much publication and trainings provided, the GIS Viewer has a regular user base of a few hundreds, similar to other major business application according to the IIS server log statistics.
- iii. The GIS Viewer development was delivered on time and within budget according to the project plan.
- iv. The Land Register system was integrated into GIS Viewer with minimum cost in 2007 even through there wasn't any extra budget for the work. According to discussion with the developer, the integration was done as part of regular maintenance work.

From the IT development point of view, the GIS integration with XML technologies was a big success too. Reasons found are as following:

- i. As explained above, the new GIS Viewer had avoided the tight coupling of the backend databases by using XML web service and XML configuration to achieve data integration. It had made the integration more flexible and robust, as the integration settings are stored in the GIS Viewer system, it can be easily

viewed and maintained. Making changes to backend databases has been easier as they no longer depend on each other with those low-level linkages.

- ii. The XML web service based functional integrations are flexible and easy to develop and maintain. According to the system design document, as long as the integration interfaces follow standards and make known to others, GIS Viewer and business systems can maintain their own interfaces without interfering each other.
- iii. The solution is configurable and open to more integration. For example it only took a short time for integrating Land Register with GIS. Documents were found for more business systems being investigated for integration with GIS in such way.

13) What are the critical success factors for the XML application(s)?

The following factors are gathered mainly from the project documents and discussions:

- i. There are powerful and mature XML and web service development tools available, such as those from Microsoft Visual Studio. Without such tools in the early stage of GIS development before 2004, the GIS team had gone straight into the backend database connection. According to the discussion records of the original development team, there was no mention of using XML web service at all.
- ii. The development team has adequate (high-level) skills and experience, with support from the management. For the second GIS integration (GIS Viewer), there were specially allocated time and resource for evaluation of different techniques and design. From the documents and history of source code, it was found the developers assigned to the project were of senior level and had a few years experience in .Net and XML with other business systems. All these can only happen with support from the management. On the other hand, when the DOCgis was developed, the GIS team used its only developer who was not familiar with .Net and other development tools, its only task was to customize the off shelf product as it was found in the discussions and source code. There was little resource available for using .Net framework and XML web services at the time.

- iii. The development followed methodologies with rigorous design and adequate techniques such as prototyping and proof of concept. As in the project plan, successful proof of concept in the early stage of GIS Viewer was a major milestone of the development. It enabled making crucial decision such as which tools or designs were chosen for the new integration system. Without rigorous processes, the development would easily end up in a wrong direction and failure.

14) What are the reasons for the XML web service integration solution being successful?

- i. XML and web service technologies have become mature and readily available for the GIS Viewer development. As in 2005 when GIS Viewer was developed, .Net development platform had been around for many years and the development team in the agency had much more experience with it from developing many systems.
- ii. The development team had gone through prototyping and running workshop to clarify integration requirements. As argued by John Crinnion (1992), prototyping was an effective development technique. From the project documents, it could be confirmed that the integration requirements were mostly based on the feedback and report from the prototype workshop. It was found the requirements document was a joint effort and authored by the business analyst, the lead developer and the business representative, the attending users have provided lot positive feedbacks for the prototype system during and after the workshop.
- iii. The documents had shown that the development team had gone through proof of concept, rigorous design and development processes. As argued by Odysseas Pentakalos (2008), proof of concept should be used for deciding development tools and final solution design. It provides assurance to the design and some guarantee to the final deliverables.

15) What are the reasons for the non-XML integration solution being not so successful?

The DOCgis is still being maintained by the GIS team and available to users from the intranet. It wasn't so successful because it failed to be adapted for GIS

integration requirements for business systems like Permissions and Land Register.

The reasons were:

- i. There was little development capability for the DOCgis at the time as there was only one developer in the GIS team with little experience in any other business system in the agency.
- ii. The DOCgis could not integrate data itself but rely only on the database interface, as it was from the off shelf product which was design for using single data source. The database linkage was tight coupling and was very fragile. According to those incidence report and discussions, there have been many incidences of system error, as any change in the business databases would easily cause break-down of the integration
- iii. The backend databases integration generated duplicated data and created many issues for the database servers. As recorded on the server system log, there have been system errors almost in daily base because problem arising from the backend databases.

A. 4 Summary of the interview with the former business sponsor

The interview has only covered a limited set of the questions from the business perspective. No technical question was asked.

- 1) What was the integration project? (How and why was it started?)

To preserve conservation value, to identify potential conflict and opportunity

To give staff easy access (tool) to GIS data for doing their jobs such as monitoring.

- 2) Why was it important to organization?

It was part of the department's mandate.

It would create new business opportunity.

- 3) What were the solutions that have been used so far? How were they achieved?

To develop a central interface joining with the current Permissions system

It should not be separate system or require software set up (installation) on user desktop.

How was it achieved?

- To start up the business initiative

- To get funding for the project
 - To draw up a project plan
 - To come up with more detail requirements
 - To involve PMU and the GIS team
 - To start the actual development
- 4) What were the major development issues and solutions for the integration project?
- Some business unit did not record any GIS info for permissions.
 - There were some politics and personality issues, in-fighting among units and within unit, hard to communicate, many conflict to resolve
 - There was a lot of GIS data to capture, have to convince conservancy IMU staff and provide revenue incentive for them to do the data capture.
 - (The actual system development was a lot faster and took much less time than I would originally thought.)
- 5) What were the critical success factors for the integration project?
- To make sure users actually use the system
 - Department's commitment to GIS and the integration system (for management support and funding)
 - Communications during the development
- 6) How are those issues managed? What were the results and reasons of them?
- To make sure the interface is user friendly and simple to use, to provide training to users.
 - To prepare a good business case to get the funding and support.
 - To have regular meetings with proper minutes and follow up actions.
 - To keep a close eye on the project management plan and budget spending.

Appendix B: Case 2 Data Collection & Preliminary Analysis

B.1 Data Source

Documentations

Documents are primary source for this case. Although DM integration is a smaller project comparing to GIS integration, but it has been developed entirely in house and has a lot of documents available for the case study. Documents for the DM integration project include the business case, user requirements, system architecture and design, functional specifications. All documents can be retrieved from various repositories within the department network.

The followings are the list of some documents reviewed for data collection:

- DM folder structure for Permissions (by DM manager)
- DM Test Scenario (by system analyst A)
- Permissions User Guide (by Permissions business representative)
- DM integration overview (by developer A)
- Land Register DM integration requirements (by system analyst B)
- Permissions II high-level requirements (by system analyst C)
- Permissions II Specifications (by system analyst C)
- Document Management Upgrade Decision (by DM manager)
- Permissions system and DM (by system trainer)
- DM Developer's Reference (from the vendor)
- AMIS to DM Interface Tips (by system analyst D)
- DM User Guide (from the vendor and DM manager)
- Various Email Messages

Source Code

There is a large amount of source code for the integration. Most of them are Asp.Net or Windows .Net projects developed in the Microsoft Visual Studio platform. They have been reviewed and listed below.

- Permissions (Asp.Net based, parts only, developed by over 4 developers)
- Land Register (Asp.Net based, parts only, by developer A)
- PermissionsDmeClient (Windows .Net, by developer B)

DMEUtility(Windows .Net, by the initial developer C and D later)

DMESmartClient (Windows .Net, by the initial developer C and D later)

DMService (Windows .Net, by developer E)

DMWebService (Asp .Net web service, by developer E)

Source code was mainly used to investigate and confirm the system design and to re-create the architecture diagrams where it is needed.

Interview

One interview was conducted with the original integration developer who is now an architect in a major IT company. His opinion on the original integration design was collected and fed into the findings and analysis. The summary of interview was attached at the end in section B.4.

System Testing

Some system testing has been done to verify findings from documents and source code. They were also used for producing screen shots for the research.

B.2 Summary of Finding

Project Summary

Since DM is the official repository of all business documents for the agency, integration requirements have arisen because users wanted to link records in business systems. The first integration project started when the Permissions system was being developed in 2002. The entire integration project can be separated into three phases. The first phase was the initial integration with the Permissions system. In this phase, Permissions system made direct API function calls to the library components (DME.EXE and CKO.DLL) came with the product from the vendor. The DME.EXE and CKO.DLL library files are installed in every client PC as part of the DM client package. For the integration, objects from the DME.EXE library were created and their methods were called for doing specific tasks, such as creating and saving documents in the DM store. The second phase happened when there was the version upgrade of DM software and new requirements for the Permissions system. Within this phase, the development team developed COM+ business server components on the IIS servers to communicate with the DM and allow IIS based business applications to provide some integration functions. COM+ components were developed in-house and installed on

the IIS server with other applications. They were built for overcome security issues. The process was complicate and required expertise in the COM+ development areas. The third phase was for incorporating Land Register system with DM, a new and generic integration solution was developed. Within this phase, the development team developed an XML web service to interface with DM and to provide most of the integration functions. The web service had not only simplified the process for the business applications to interact with DM, but it had also allowed more flexibility in configurations, such as talking to multiple DM stores. So far Permissions and Lang Register systems both have integration features with DM. There are plans for integrating more business systems in future.

Phase 1

According to the user guide and business requirement document, the integration functions were:

- To allow user to use Permissions data to create Word documents and save them into DM within the Permissions system.
- To create and display hyper links for DM documents in the Permissions system, so users can click and open up DM documents in new windows from Permissions.

Most of the integration work was concentrated on the first part of the requirements, which was to create Word documents from within the Permissions system and save them to DM. The initial plan was to build the function as part of Permissions system within the ASP.Net application and have it running on the IIS server. But when the development team tried to activate Word on the IIS server and to pass the current user credential to the DM server from the IIS server, it encountered some irresolvable problems. User's account on the browser PC could not be passed to the DM server. After much investigation and experiment, the project team had to adopt a more complex solution based on the old ActiveX control embedded in the web page working on the client PC. A purpose built ActiveX control was used to start up Word, create the document and save it to DM using the current user credential from the browser PC. After the new document has been created, the ActiveX control would then sent back the new document detail including the ID to the web server for creating a new link record between the document and the Permissions record.

The followings are screen shots for the document creation process:

1) *Permissions screen of document creation from available templates*

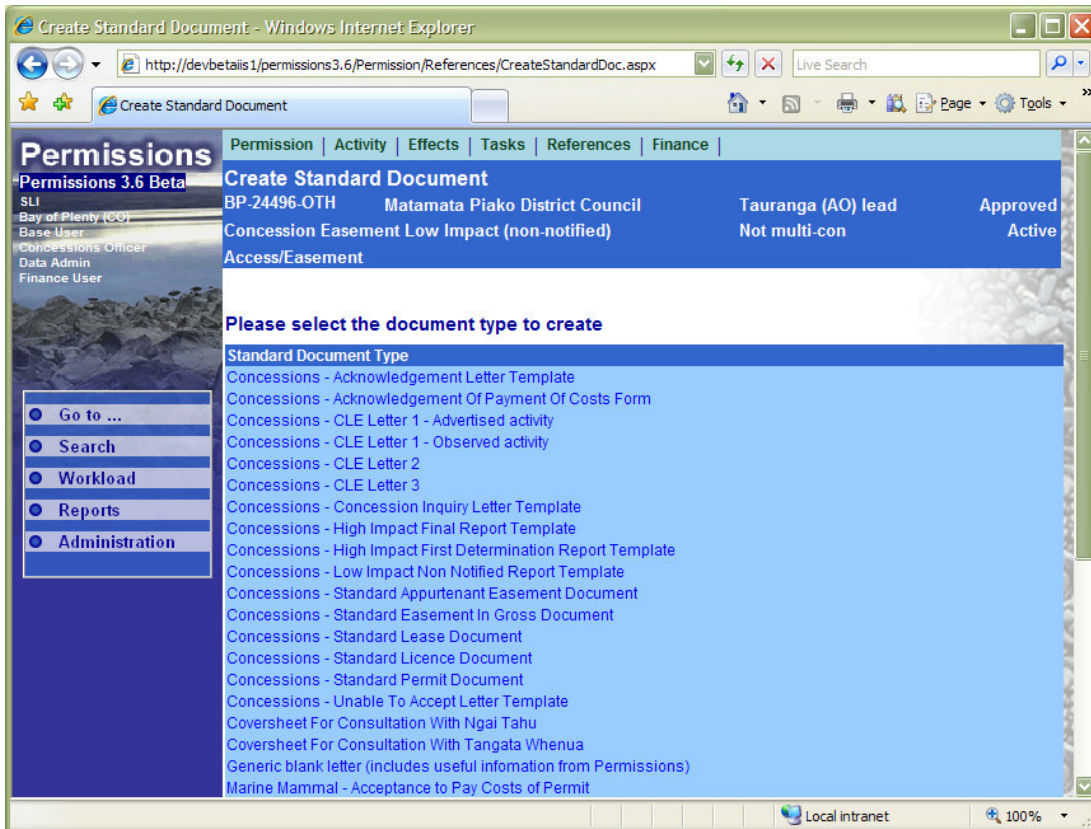


Figure B. 1 Document Creation Screen 1

2) *Screen created by the ActiveX control to confirm the document creation*

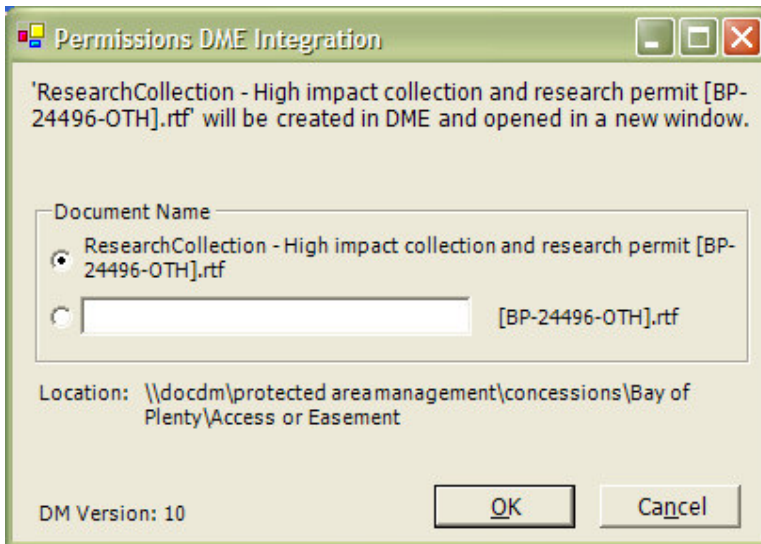


Figure B. 2 Document Creation Screen 2

3) Example of Word document screen populated with sample Permissions data

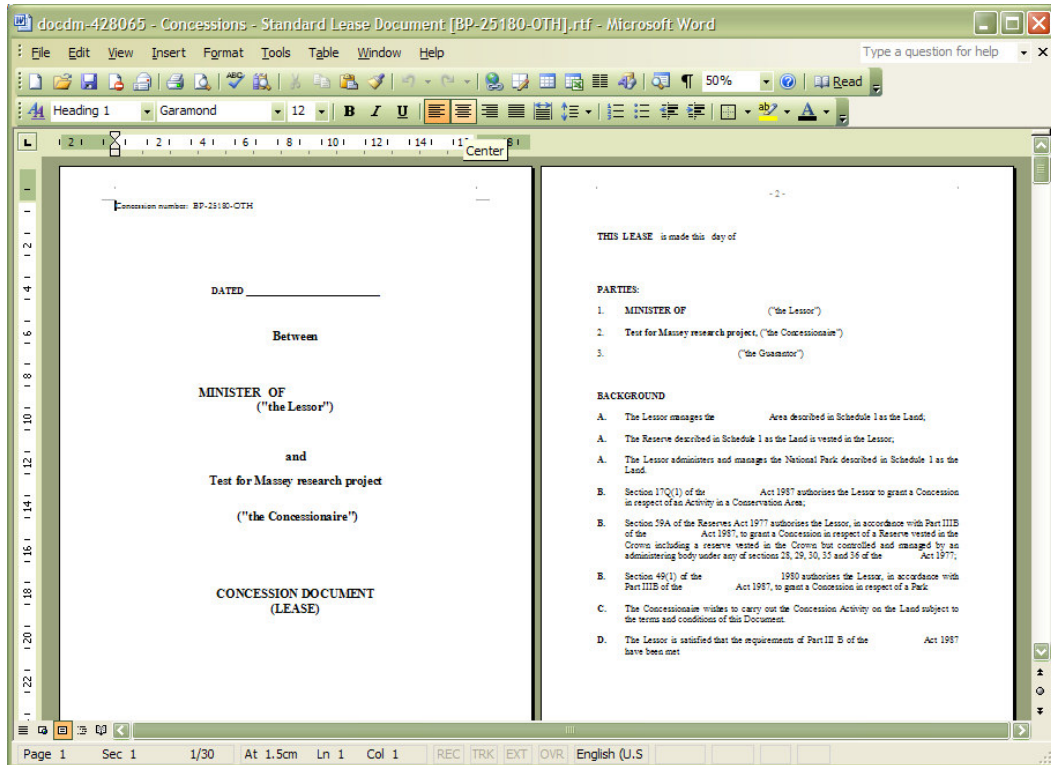


Figure B. 3 Document Creation Screen 3

The second part of the requirement is relatively simple, because DM already has build-in feature for taking in a document ID as part of URL address and open up the document. Permissions system displays DM document IDs such as DOCDM-394555 as a hyperlink, when the user click on it, it would open up a new browser window and pass “dme:// DOCDM-394555” as the target URL. The new window would then submit the request to the DM server, and open up the document with ID “DOCDM-394555” in its original format such as Word or Excel using the respective Office product.

Example of Permission screen with linked documents

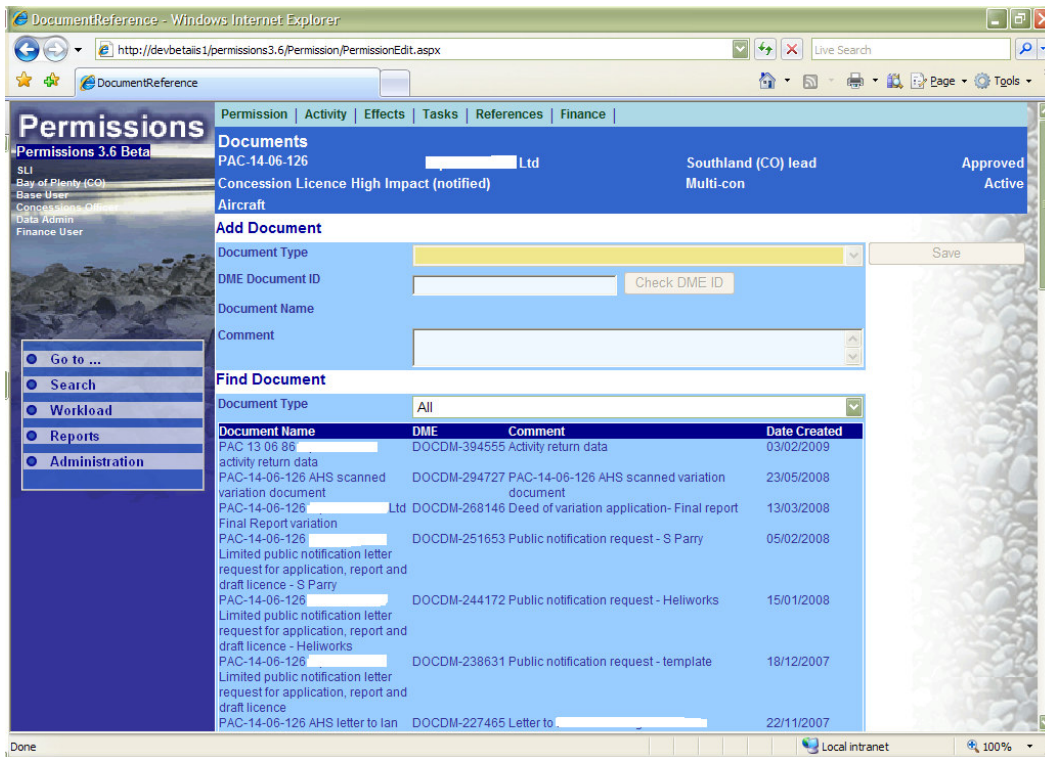


Figure B. 4 Document Integration Screen

The following architecture diagrams have been re-created with information from the design document and source code, including all the ASP .Net apps and DLL components.

Architecture Diagram for the Part 1 Integration - Document creation within Permissions

(Re-created from the source code)

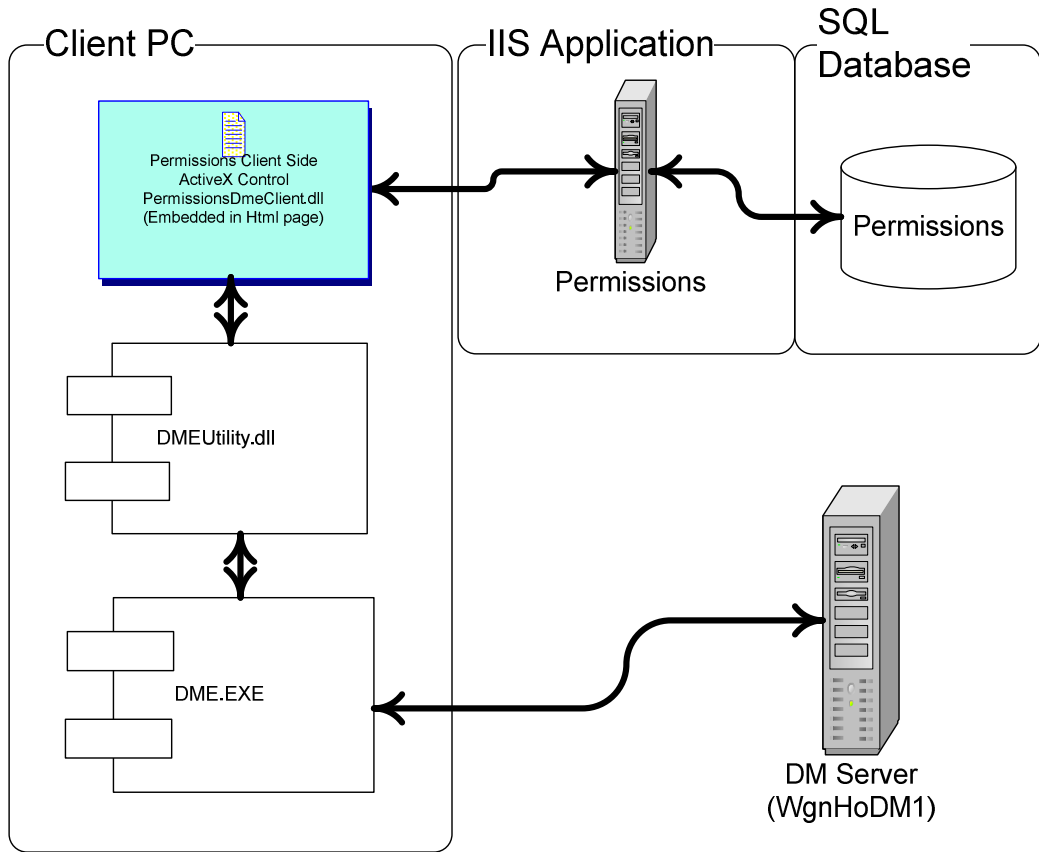


Figure B. 5 DM Integration Architecture 1

Architecture Diagram for the Part 2 Integration - Display of document hyper links within Permissions

(Re-created from the source code)

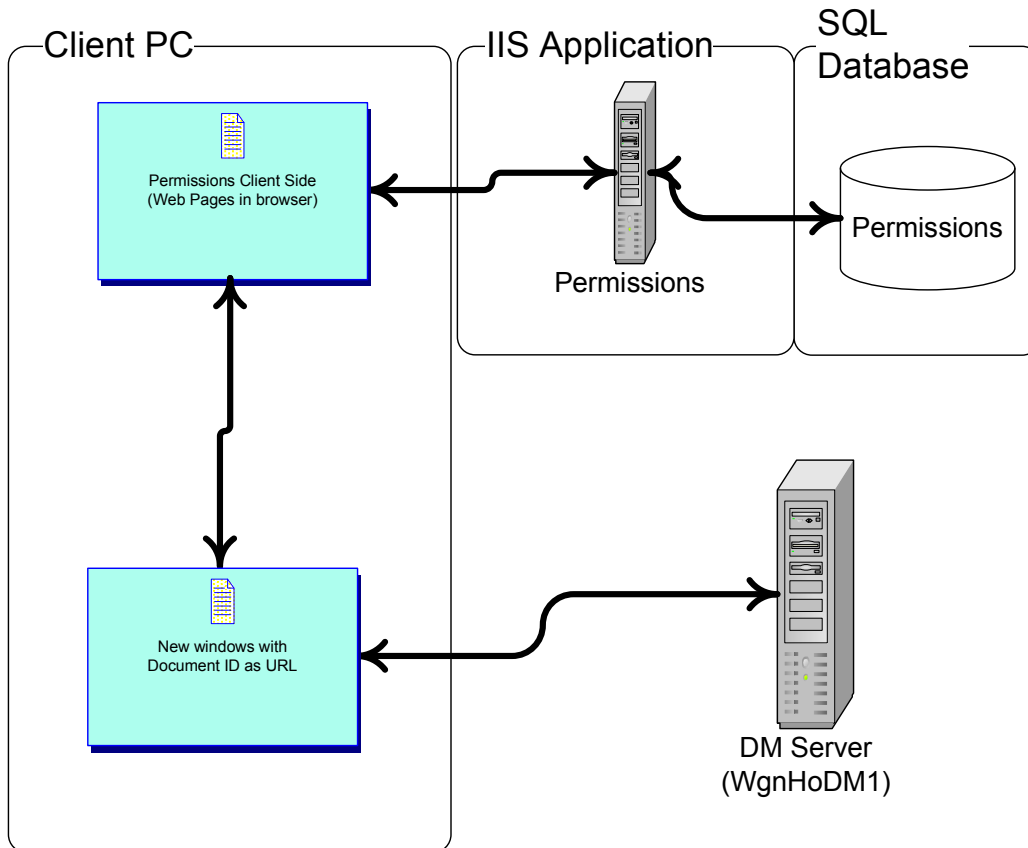


Figure B. 6 DM Integration Architecture 2

Phase 2

In 2004, the department proceeded to upgrade the document management system from version 6.5 to version 9. In the same period of time, there was an enhancement request for Permissions systems to allow users to dynamically check and edit the linkage to the DM for those linked documents details.

According to the project plan and requirement document, there were two main parts of the work. First, because the upgrade of DM was rolled out across the country conservancy by conservancy, the client side ActiveX component of document creation had to be rewritten to cater for different versions of DM software installed in the local PCs. Also, with the new DME.exe, the ActiveX component was no longer able to work within the IE browser's process space, a new EXE process had to be used to create the

document and communicate to the Permissions server. A new web service interface was also developed for saving details of newly created documents to the back end database. It replaced an existing web page that generated no web page as response but was only for processing posted data from the browser. The second part was for the enhancement of dynamic query to DM. It required the Permissions application on the IIS server to communicate with the DM server directly, not through the client side PC where the browsers reside. Permissions system is an ASP.Net web application, most of its process is on the IIS server side, such as querying the databases and generating the web pages. The new function would be built and added to the existing system.

According to the design document, the solution included installing the necessary client component DME.EXE on the IIS server and using it to talk to the DM server. For retrieving document attribute data from the DM server, the Permissions system had to use a COM+ component to call DME.exe library functions to communication with the DM server. That was because the DM server only allows queries from users with network credential. The design was to build and set up a COM+ application to run with a network credential for doing work. COM+ applications are common form of server applications hosted by Windows.

There weren't many changes in the Permissions screens in this phase of the integration as explained above. The bulk of work was done behind the scene, particular in system design and architecture. The bulk of the source code had been modified too. The new architecture diagrams are re-created from the system design and source code as followings:

Architecture Diagram for Revised Part 1 Integration - Document creation

(Re-created from the source code)

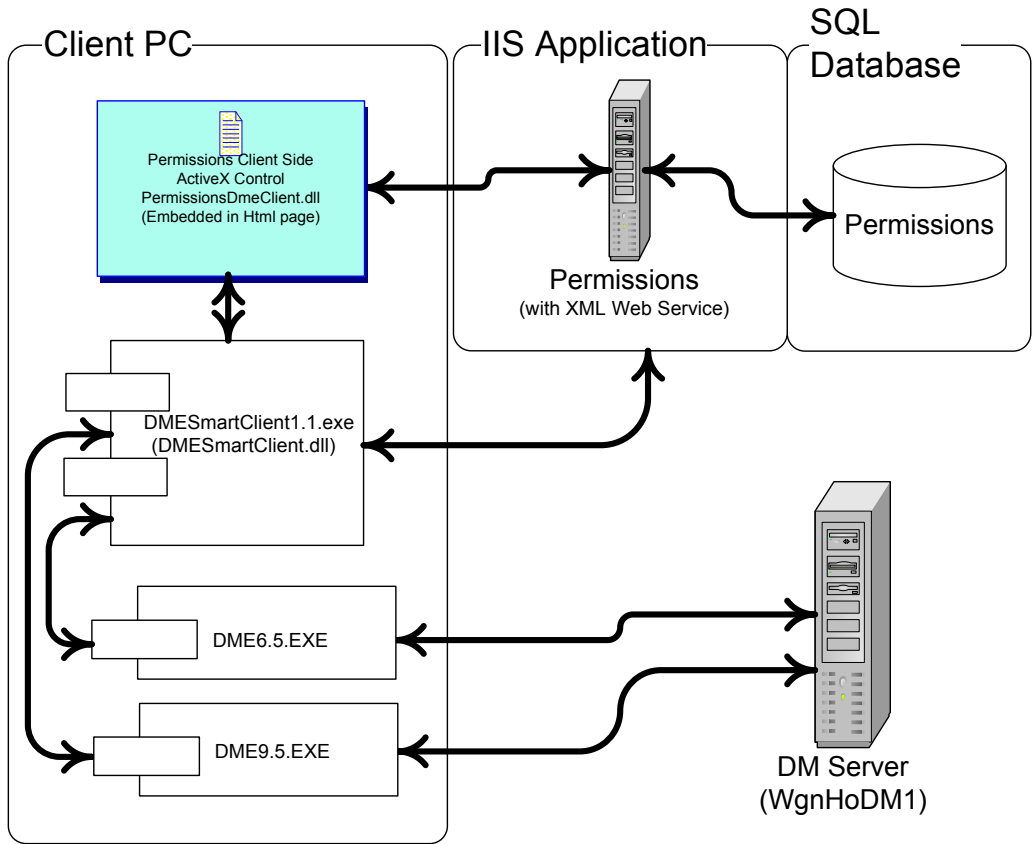


Figure B. 7 DM Integration Revised Architecture 1

Architecture Diagram for Revised Part 2 Integration - Dynamic DM querying

(Re-created from the source code)

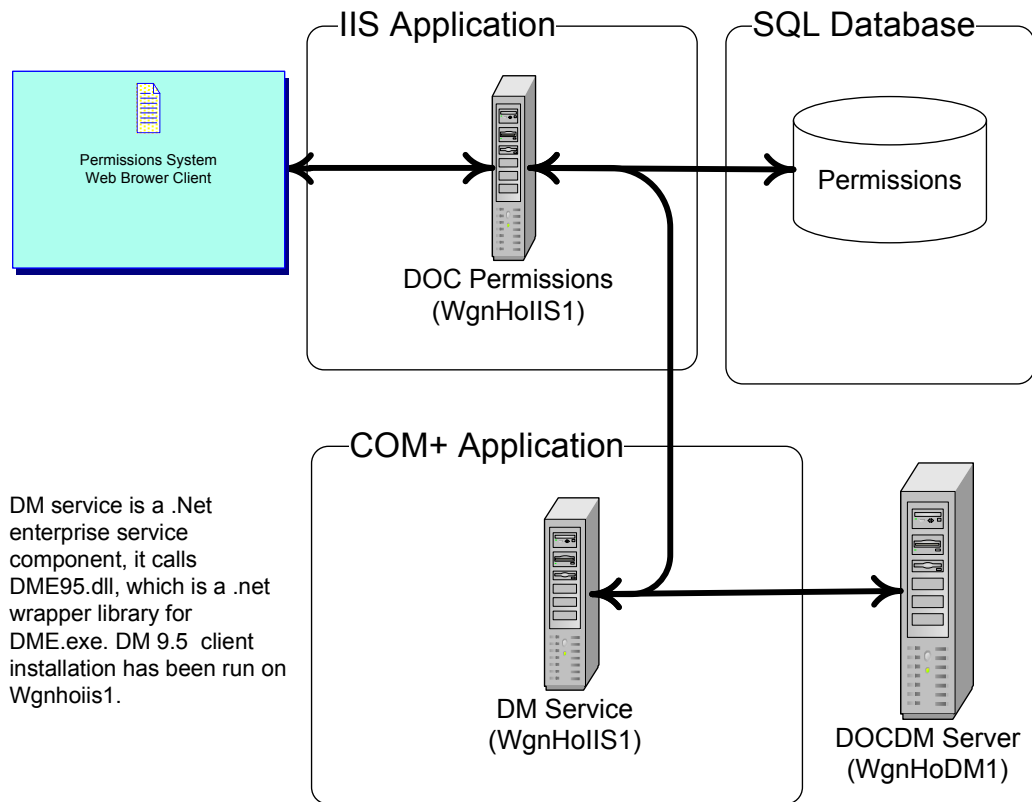


Figure B. 8 DM Integration Revised Architecture 2

Phase 3

In 2006, the department again proceeded to upgrade its DM system from version 9 to 10, which was a major change on the server side from using scattered MS Exchange servers to a central MS SQL server for document storage. The client side document creation component was re-written again to cater for the newer version of the DM software installed in the local PC, in a similar fashion as it was done in phase 2. The component was re-developed using a software design pattern to cater for multiple versions of server components. The new design would reduce changes in the source code for any future version upgrades.

At the same time, another business system Land Register was to be integrated with DM in similar fashion as Permissions, but without the document creation function. Although both Permission and Land Register were developed in ASP .Net, but they used different version of .Net framework, which caused the COM+ component which

was built in one version of the .Net framework can not be shared between them. In the end, the solution was to build an XML Web service to provide the integration functions, which had constraint on calling clients as long as they understand XML web service. The new XML web service not only can be used by the Land Register system without creating duplicated COM+ component, but it also opened up possibilities for other business systems to be integrated with DM in future.

There were some other issues to be addressed in the time for the integration. According to discussions and email document, there had been some performance issues with the existing COM+ components running on the IIS server for the integration, typically they were resource hungry and tended to run very slow. The development team had consulted with the DM software vendor in Australia; it recommended that a new library CKO.DLL should be used instead of the 'old' DME.EXE on the IIS server by the COM+ component for querying the DM server.

Again, most of the work for this phase was done on the backend IIS server, particularly on the components for querying the DM server. The changes were also identified in the design document and source code. They are illustrated in the following diagrams.

Architecture Diagram for revised Part 2 Integration

(Re-created from the source code)

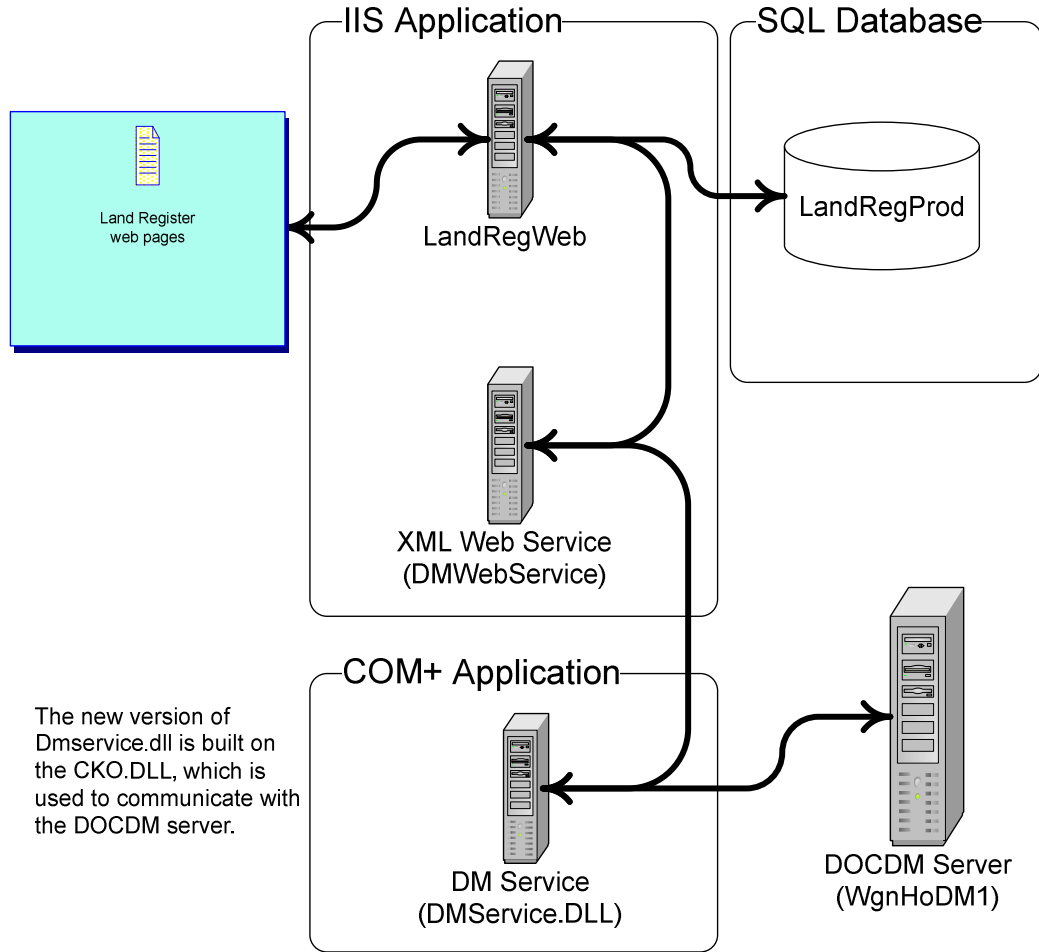


Figure B. 9 DM Integration Second Revised Architecture 2

Example of Land Register screen with documents linkage information

Land Register v1.2																										
Search	Report	New	Delete	Save	Utility																					
Cons Unit	Reserve	Related CU	Leg Auth	Parcel	Admin Unit	Val Ref	Map Ref	Permissions	Documents																	
G42114: Obelisk Creek Conservation Covenant [RACC, P]						Excel	History	Archive	Transfer	Print																
Document List																										
<table border="1"> <thead> <tr> <th>DOCMD_No</th> <th>Title</th> <th>Created Date</th> <th>Modified Date</th> </tr> </thead> <tbody> <tr> <td>DOCMD-267383</td> <td>Site inspection - Feb 08</td> <td>11/03/2008</td> <td>12/03/2008</td> </tr> <tr> <td>DOCMD-180099</td> <td>Obelisk_covenant_doc700</td> <td>3/08/2007</td> <td>3/08/2007</td> </tr> <tr> <td>DOCMD-267590</td> <td>Obelisk Covenant SO</td> <td>12/03/2008</td> <td>12/03/2008</td> </tr> </tbody> </table>											DOCMD_No	Title	Created Date	Modified Date	DOCMD-267383	Site inspection - Feb 08	11/03/2008	12/03/2008	DOCMD-180099	Obelisk_covenant_doc700	3/08/2007	3/08/2007	DOCMD-267590	Obelisk Covenant SO	12/03/2008	12/03/2008
DOCMD_No	Title	Created Date	Modified Date																							
DOCMD-267383	Site inspection - Feb 08	11/03/2008	12/03/2008																							
DOCMD-180099	Obelisk_covenant_doc700	3/08/2007	3/08/2007																							
DOCMD-267590	Obelisk Covenant SO	12/03/2008	12/03/2008																							

Figure B. 10 DM Integration Land Register Sample Screen

B.3 Preliminary Analysis

Apart from the general data analysis methods of this research thesis as they were mentioned in the previous section, there were some particular aspects for the DM integration project as followings:

- 1) There were a large amount of source code for the different phases of the integration, the research had to go through them and re-created the architecture diagrams.
- 2) The source code was also used to establish chain of evidence in many areas, such as the integration design and their advantages and disadvantages.
- 3) Comparisons including counterfactual comparisons have been made in many areas for different integration approaches.

Case Study Questions:

1) What is the DM integration project in different prospective?

From the business point of view:

By investigating into the DM product and the department business, it was found that Document Management is the central repository for the agency's documents. Because it is such a large organization, there is a sheer volume of documents it is required to manage. Within the department organization structure, there is a "Knowledge Service" team of 13 people responsible for looking after the library of the agency including DM system. It had been a huge task for the software roll-out, hundreds of other documents and email records were found related to the project including implementation, system features and user training. The DM integration is important because it makes the DM accessible from existing business systems, which not only extends the DM accessibility but ensure the integrated business systems make use of DM for their documents. The integration would streamline the business processes and improve productivity in document creation and maintenance for those business systems.

From the "Permissions User Guide" document, the DM integration features are summarized as followings:

"The Permissions system can produce many of the SOP template based documents for you with information filled in, and also keeps track of DME

numbers related to the Permission with hyperlinks that you can click on from the permission record to open your documents.”

In details, there are two main features of the integration:

Creating Standard Documents

Permissions includes a function where SOP template based documents can be automatically generated from the SOP templates with information about the Permission and Holder (or applicant) automatically filled in.

Linked DM Documents

Any document stored in the Department’s DM can be linked to a Permission record so you can easily find the documents. This feature can be used for applications (if they were received electronically), reports, permission documents, monitoring reports, cost estimates, etc.

From the development point of view:

Both DM and Permissions are core business systems for the department. However, they are of complete different software, one is a third party Windows base client-server application, the other is an in-house ASP .Net IIS web based application.

2) *What are the business goals for the project?*

From the business point of view:

As summarized in the Permissions requirement document, the direct goals of the DM integration were:

- i. To allow Permissions users to create documents including legal contracts using pre-defined templates from within Permissions system.
- ii. To allow Permissions users to link existing documents with records in Permissions.
- iii. To allow Permissions users to open relevant (linked) document from the Permissions system through hyperlinks.

These goals had been reflected and achieved by the system design and necessary software components as the investigated into the source code and other documents.

From the system development point of view:

The project had been a challenge for system integration in the agency because of the major difference of the sub-system, the development goals have changed through the different phases of the project, as described in the findings section.

- i. In phase 1, it was to develop a workable function to allow Permissions users to create documents including legal contracts using pre-defined templates from within Permissions system.
- ii. In phase 2, it was to modify and improve the solution so it could work with the changes of the DM software upgrade.
- iii. In phase 3, it was to further modify and improve the solution so it could work with the changes of the DM and new integration with other business system.

3) *Why is it important to the agency?*

DM is the central document repository for the agency. The use of DM and its integration with business systems like Permissions and Land Register are important for the department in many ways:

- i. DM is an important piece of software as explained above, as document management software, DM makes it easy to store and share document within the department. It provides many extra document management abilities, as they are described in the “DM User Guide” for the agency.
- ii. As stated in the business requirement documents, through the integration, business systems like Permissions allow users to take advantages DM in the simplest method, such as a single click to create and open a document from DM. According to the discussions with the original trainer, who had gone around the country to train local users for the system, Permissions users were very impressed with the effectiveness of the function and its easiness of use. As explained in the finding section above, the integration would complete the document creation task with two mouse-clicks. One for selecting a document, one for confirming the action and the name of the new document.
- iii. As stated in the Permissions User Guide, the integration improves the productivity and accuracy of the document creation. This was because the new document will be filled with data automatically from the database. These data include permission holder name, term start date, end date and etc. The process likes an ah-hoc mail merge with even less effort.

4) What are the solutions that have been used so far?

By reviewing all source code and design document, the architecture diagrams have been created and displayed in the previous findings section. They are summarized in the following table:

Core Components	Core Technologies
DM library component DME.EXE and CKO.DLL	Windows COM
PermissionsDMEClient.dll	Windows COM
DMESmartClient.EXE	Microsoft .Net
Interface components Permissions and Land Register systems	XML and ASP.Net Microsoft .Net, SQL, IIS server
DM service	COM+
DM Web Service	XML web service

5) What are the critical success factors for the integration project?

i. Clear Business requirements

They are fundamental for system developments, and it was the same with DM integration. In the “Permissions User Guide” document, the integration features are clearly described, although they are relatively simple comparing with other systems, such as GIS integration.

ii. Suitable software and development tools and solution design

The most complicate part of DM integration lies in the solution to connect the different technologies that were used to by DM and business systems. For the initial integration, as original developer said, “Many other possible approaches had been tried before the design was settled on the .Net ActiveX control.” The function was innovative one that had not been done before in the department’s systems. Under pressure for delivery, the developer had created the solution that it wasn’t the best he would have wanted as he suggested in the comments. By going through the history of the integration development in the documents and source code, the researcher found there had been major changes to the integration and development tools. For example, the DM software version had been changed from 6.5 to 9, then from 9 to 10. Most document creation components for the client side had been

rewritten and changed substantially. The .Net version for business systems had also been changed from 1.0 to 1.1, and from 1.1 to 2.0. All these changes were fundamental changes and had affected the integrations.

iii. Projects management

Project management is always crucial and the DM integration seems managed quite well, although it was in strict timeframe for the initial integration, which was reflected in discussion with the developer. In the following phases, developers were able to try out options for finding the best possible solution. There was specially allocated time and budget for the DM upgrade as they were shown in the project plan.

iv. User Acceptance/Training

There was special user training program run by dedicated IT trainer in the agency for the Permissions system. The DM integration was included into the training package. The function was also clearly documented in the Permissions User's Guide document. Adequate user training was an important factor for the function will be widely used, as near 20,000 records had been created since.

6) *What are issues and challenges faced by the project team?*

i. The third party software was built in old technologies.

DM is a product of an Australia company with a relative long history. The product was build as traditional Windows client server application. Integration with such application would only been done through strict interfaces exposed in the library file. Most of the agency's in-house applications such as Permissions were built in the latest .Net platform. The development had encountered majors issues in integrate DM through its library component.

ii. Support was very limited from the DM vendor.

Although there is a "Developer Reference" from the vendor; there was little other technical support from the vendor as it was recorded by the development team. The vendor only has an agent in Auckland, which only acted like a middleman and passing messages only. By looking through the email communication records, it was found it took many forward and backward emails to convey single message. In some of these emails, non-

technical issues like service delivery agreement had got into the way many times. As recorded in various email messages, the development team had many attempts to try to get some explanation and support from the vendor through its agent. From time to time, other issues like service contract agreement got into the way and prevented direct and useful communication to the vendor. Eventually the development team managed to have one teleconference with the vendor's technical staff. During which they recommend the agency to used CKO.dll on the IIS server.

- iii. The initial development had only short time frame.

The initial solution was developed under pressure of delivery deadline, as original developer said, "I wished there would be a bit more time for the development at the time". As results, the design was not well thought for future changes and enhancement. By going through the changes in the source code, it can be concluded the integration components (DLL and EXE) had been through many changes. As the initial developer mentioned in a discussion, he was not able to develop the integration component in the way he thought was the best approach, he had just enough time to get the document creation ActiveX component works at the time. He said he would have done a much better job if he had got a bit more time. The component has been changed substantially since as all changes had been identified in the source code during the research.

- iv. Software components have changed over time, including all sub-systems. The sub-systems of the integration have changed over time. As it can be identified in many documents, DM has changed from version 6.5 to version 9, and then to version 10. The Microsoft .Net platform has also change as from version 1.1 to version 2.0. The Permissions and Land Register systems were built in different stages and require different version of .net framework to run.

- v. When there was a run-time problem in production, it was difficult to track down.

There have been some run-time issues encountered in production only for the COM+ component, they were hard to track down and resolve. The DM Service COM+ component is installed on the IIS server, as for production server, the development team has no access at all, which made it very hard to investigate any run-time problem. The issue was raised in a couple of defect

reports; it became one of the reasons for the change to using another DM library and XML web service for the integration. According to the department deployment process, only the production support team has access to production and is responsible for copying files package prepared by development to production server.

7) *What are the major technical issues and solutions for the integration project?*

- i. The third party software was built in old technologies.

It was a big technical challenge as mentioned above. There was only one technical reference document for this integration. According to the vendor's "DM Developer Reference", the document was a complete guide for development using the DM component library, such as integration with MS Word. As the developer recorded, the reference document was very useful and was the only technical document he used for the initial integration development. There are no other usage of the DM library could be found in the source code repository for any other systems except this integration. A search on the Internet with key word "DME.EXE" would return nothing useful. Any other technical reference would be very limited if they are exist at all. In the later phases, the development had allowed special time for testing the library component as it was recorded in the project plan. However, issues still appeared when the library was installed on the server. The component did not perform well as it was reported by the production support team. There were multiple instance of DME.exe process running on the server and the system was very slow to respond at the time. It was then later confirmed by the vendor that the DME.exe was not designed to run on a server but only for a client PC. The integration was then changed to use a new component CKO.DLL as recommended by the vendor.

- ii. Software components have changed over time.

As documented, there have been three versions of the DM and its library components, which the integration had to use. All these changes required re-work or re-development on the integration software. In the second phase, the investigator had identified software patterns were used in the source code for enabling the integration software to adopt to work with different versions of

components easier. A COM+ component can only be used by a system built in the same .Net version, but the Permissions and Land Register are of different .net version. It would not be desirable to have duplicated COM+ applications with different .Net version, as it there would be a maintenance burden. In the last phase, XML web service was adopted to allow business systems to use the integration component regardless of their .Net version.

- iii. When there was a run-time problem in production, it was difficult to track down.

Because of the rules safe guarding the production server, developers have no access to the production and it makes it very difficult for developers to track down and fix production problems. In the last phase, the XML web service was adopted also for wrapping up the COM+ application, as the XML web service could be designed to better handle the COM+ application and any error raised from it. The COM+ application was also changed to use a new library as recommended by the vendor.

8) *How has XML been used and to what degree?*

By going through the design document and source code, a few XML applications have been identified and described as following:

- i. In phase 2, a XML web service was added for the integration. The web service (called DocumentService) is resided within the Permissions IIS application. It is a replacement to a previous web page (called StandardDocumentHandler.aspx), which was used to process data sent from the client side and to add new document linkage records to the backend database.
- ii. In phase 3, another XML web service was adopted for providing an interface for new business system to be integrated with DM. The XML web service (DMWebService) was built for allowing the Land Register system to used the COM+ component (DMService) for communicating with the DM server, as Lang Register and the COM+ component are of different .Net version and could not communicate directly. The web service was a simple wrapper with less than 100 lines of code statements.

9) What are the reasons for not adopting XML in the projects?

When the integration was originally developed, XML was not adopted in areas where it would be possible, as described below:

- i. The document creation could only use client side ActiveX control.
From discussion with the developer in the initial development, there was an attempt for getting the document creation work on the server would lead to using XML web service, but it did not succeed because he could not make the document created using the user account on the browser side. He has not enough time to investigate further. It was concluded that document would only be created on client side through the DME.EXE library provided by the vendor. According to developer reference document, the DME.EXE is a COM component library installed as part of the DM client PC set-up.
- ii. The saving of document linkage function was not designed properly.
In the first phase, the developer had commented, XML would have been used for saving new linkage records, but as the work was done in a hurry, a ‘quick-and-easy’ web page was created for doing the work. As the original developer said, “There was not enough time to convince the lead developer to use web service approach. It was a new concept for her and would take time for her to accept it.”
- iii. The DM server could interpret document hyperlink URL.
As it was documented in the DM user guide, DM has built-in features of using URL to access documents over the network, such as by passing “dme://docdm-12345” to a new browser window. There wasn’t any need for extra work for enabling the hyperlinks to work for the integration. Hyperlinks are just made up and embedded in the Permissions pages. As shown in the previous finding section, by simply clicking on the document ID, the document will be opened in a new window. As a result, no extra interface such as XML web service was needed.

10) What are the reasons for adopting XML in the projects?

- i. XML web service for saving document linkage data into the back-end database
By looking into the design and source code, the initial web page (StandardDocumentHandler.aspx) for saving document link record did not

generate any html result and return it to the browser. It was just for taking in posted data and save it to the database. The design was a quick and dirty approach and it is not what web page should be used for. According to the Microsoft ASP .Net training guide, asp .net pages (aspx) are technologies for creating user interfaces, not for data processing. The XML web service was a much better way to received data from the client side, process them or save them in the database server. As defined by the W3C, web service is "a software system designed to support interoperable machine-to-machine interaction over a network", which is quite general, but as argued by M. Hansen, S. Madnick, and M. Siegel (2002), as web services standardized the programmatic interfaces for application-to-application communication, it increased the flexibility of data processing and integration. With the Microsoft .Net framework, the XML web service was easy to build and deploy, it is much better option than others, such as using web page posting (including aspx), direct access to database server or by COM+ component. In summary, XML web service was better option for data processing.

ii. XML web service for sharing integration function

As shown in the design and other documents, the XML web service (DMWebService) was built in phase three for overcoming the compatibility problem caused by different .Net version used by Land Register system and the COM+ component (DMService) for communicating with the DM server, i.e. they could not communicate with each other directly. Because XML web service uses open standards and protocols, such as XML, SOAP, WSDL and UDDI, it can be shared by any business systems as long as they understand web service, regardless of the platform, such as the .Net version they were built on. The web service was a simple wrapper, it was a much more effective way than re-creating the COM+ component in a different .Net version. If a duplicated COM+ component was created, it would not only difficult to deploy and confusing to others, it would create unnecessary maintenance work. In summary, XML web service would be used by systems regardless their software platform or their version.

11) How were the decisions of the system design made?

No document has been found to give direct explanation of the decision making process for the project. By going through the documents and source code, summarizing the changes, the followings are some findings:

- i. The original integration was done by a contracted developer under pressure to just get it to work.

By checking the source code of the original release of the Permissions system including the DM integration function, it was found there were two contributing developers. One was the lead developer, the other was the contractor. From discussions with the contract developer who was mainly responsible for the integration functions, he said that he was under pressure to deliver the system. The lead developer's expectation was to have a quick delivery first and to fix things up later. He felt he was not able to deliver the best solution under such short timeframe. He had let the project manager aware but the manager had decided to go with the lead developer's opinion at the time, although in the later phase, the manager had made sure to allocate enough time to deal with the DM integration, as the project plan had shown.

- ii. The development later had included improvement work on the original design.

There have been a number of changes since the original release, as they are described in previous sections and identified in many document and source code. A lot of the changes were for improving the integration, such as adding software design pattern and XML web service. The decisions for those changes were based on better knowledge of the integration techniques and better development resource at the time. For example, as explained in previous analysis, XML web service is one of the latest techniques for system integration with many unique advantages, its application for the integration had not only resolved problems such as different .Net version issue, but also improved the design, such as replacing the data processing web page. Another reason for changes to XML web service would also be that the .Net development tools have been getting more mature since the integration was original built in 2002; the development team has accumulated more skills and experience. During the same period, many other XML web service have been developed, including those for the GIS integration.

- iii. Future consideration was also included in the latest work for the integration. When the Land Register system was included in the DM integration, it created some new challenges and also made the development team to put more consideration for future integration in the system design. In the business requirement for the Land Register DM integration, Land Register users wanted the similar DM functions in the Permissions system; it is likely that other systems would follow in the future. In the system design, reflected in the architecture diagram, the XML web service (DMWebService) for communicating with the DM server was the result of such thinking. By making the integration function platform and language independent, it is open for more future systems.

12) How successful were XML web service integration solutions when XML was actually used?

The XML web services in DM integration were successful. The reasons were identified as followings:

- i. The DM integration is one of the core functions in the Permissions system and has been running successfully since its original release in 2002. By querying the database, it was found that over 18, 000 DM document having been created or linked to Permissions so far. No error or defect as been reported from the system was particularly related to DM document creation function, although there were many others for other part of the system. As it was commented by the trainer, the DM integration function was well received by the users.
- ii. As shown in the requirement document and work estimation, the DM integration with Land Register was implemented with a small budget and in a short timeframe. By using XML web service to create the DMWebService, which provided most of the necessary integration functions, the Land Register system only need to interface with web service, which the system had already done in many other areas, such as GIS integration. Land Register also used XML web service to access its database as the research found by looking into its design and source code. The DMWebService was built in very short time, because it is a simple wrapper on the existing COM+ component, as it was shown in the system design and architecture diagram.

- iii. The XML web service is extendable and is ready to be used by any future business systems integration with DM. According to the system design, with the XML web service interface between the business systems and DM, any future change on the DM server can be handled within the web service functions without affecting the integrated business systems such as Permissions and Land Register. XML web service is also platform independent; it would be used by any business system as long as it uses XML and SOAP. The integration with Land Register was a good example in such situation, as it has a different .net version from the original COM+ component used for the integration.

13) What are the critical success factors for the XML application(s)?

- i. Integration with third party software required more skills and effort. As DM is third-party software, it was complex and its design and source code are not accessible by the integration developer. The only useful document found during the research was the developer reference. It was more difficult to work with than other in-house systems. As commented by the initial developer, he would need more time to develop a satisfactory solution.
- ii. System design should follow best practice. Because in the initial phase, the integration was done in a ‘quick and dirty’ approach under delivery pressure, as recorded by the developer, it left some issues for later phase to fix, including the misuse of aspx page for data processing. In the later phases, more robust design was implemented using the XML web server, including the web service for data processing and communicating with DM server. The large amount of changes found in the system design and source code indicated a large amount of work involved, some of them would have been unnecessary if best practice were adopted in the first place.

14) What are the reasons for the non-XML solution being not so successful?

Part of the integration solution including those non-XML components was not so successful, because

- i. In the initial phase, a web page was used to process data instead of XML web service, it was a ‘quick-and-dirty’ approach and XML web service would be a

much better solution. As defined by W3C, web service is design to support machine-to-machine interaction over a network. Web service is also recommended by Microsoft as best practice for pass data between applications. As shown in the source code, the issue was rectified in the later phased of the integration.

- ii. Using client side ActiveX control to create Word document, as commented by the original developer, was not the best way it would be developed. With the large amount of the changes and work involved in the later phases, it is proven to be a solution with high maintenance cost. The design nature of DM product would also be the main contributor, from the email records; the vendor would not recommend creating documents on the IIS server. By looking into the source code, it was found although the current solution had been improved a lot from the original design; it still has to use client side components on the browser PC.

15) What are the reasons for the XML web service integration solution being successful?

The XML part of the DM integration solution was successful, because

- i. The design followed the best practice.
As explained above, XML web service was designed for machine-to-machine interaction including passing data for processing. The later development was done to make the system design more robust, because the initial development lacked the time for doing so as recorded in the documents from the initial phase.
- ii. The XML web service is platform independent.
As defined by W3c, XML web service uses SOAP protocol, as long as the application understand the protocol, the application can be from any platform, such as Java, and .Net, regardless of the different version of .Net, as they all use SOAP. As it was illustrated in the architecture diagram for the phase three development, the XML web service was created for overcoming the different .net versions problem. ,

B.4 Summary of the interview with the former developer

- 1) What is the integration project?
To integrate the DM and the Permissions system.
- 2) What are the goals for the project?
To provide a seamless integration of DM and the business system.
- 3) Why is it important to organization?
It provides better user experience. The data will be more accurate. It is a lot more efficient than to use the two systems separately and manually.
- 4) What are the solutions that have been used so far?
One click on the button to start and complete the document creation and saving data to the database.
- 5) What are the critical success factors for the integration project?
Performance, user experience, data security and integrity Project management. Management support. It was the first time integration of DM with business system. Understanding DM (API s) and .Net
- 6) What are issues and challenges faced by the project team?
The project was a bit behind. I wished there were a bit more time. The technology was new and more research was needed.
- 7) What are the major technical issues and solutions for the integration project?
 - DME was a COM based legacy system. Special technique had to be used. The main system was a .net web based system. We tried creating the document on the server side, but in the end, we had to adopt the client side approach. It was a challenge.
 - Another challenge was to save data on the server, as web service was new at the time; we used the traditional aspx page to do the job. But I think web service approach was a much better way.

8) How has XML been involved in past and current system integrations?

We didn't use XML in the beginning.

9) What are the reasons for adopting or not adopting XML in the projects or in the project past?

There wasn't enough time for doing research. The people in charge of the development at the time did not know enough of the technology. I believe XML web service would be more robust.

10) How were system design decisions made?

For document creation, we did a lot of research, but there was a big constraint with the DM API. We did not have many options.

For saving the data using the aspx page, we did so because we did not have much experience in web service at the time.

11) What are the critical success factors for the XML application(s)?

- To understand the technology well.
- To have common understanding and buy in from the whole team.

12) What are the reasons for XML web service integration solutions being successful / not successful?

It needed a change of mindset from traditional API to SOA approach.

Appendix C: Case 3 Data Collection & Preliminary Analysis

C.1 Data Sources

Documentation

The office and person data-sharing project is also a very long running integration project. Many people have been involved and left (the department). There are a large number of documents for the project, including the business case, user requirements, system architecture and design, functional specifications and even the full set of source code.

The followings are the list of documents reviewed for data collection.

- Person & Organization (PerOrg) information system proposal (by system analyst A)
- BioWeb PerOrg Prototype Screen Dumps (by developer A)
- PerOrg inetOrgPerson Object Class (by system analyst B)
- PerOrg User Requirements (by system analyst C)
- PerOrg Vision and Scope (by system analyst C)
- PerOrg Project Term of Reference (by system analyst C)
- PerOrg Software Requirements (by system analyst C)
- PerOrg Presentation (by developer B)
- Office and Person Issue Discussions (by system analyst D)
- Functional Specification - Permissions - Middle Earth Interface (by system analyst E)

Source Code

Database Objects:

The followings are objects from the old database by the former DBA A, which are to be phased out:

- CommonDB (database)*
 - OfficeMap (table)*
- DOC(database)*
 - Office (table)*
 - OfficeType (table)*
 - Person (table)*
 - Role (table)*

PersonRole (table)

PersonMenuGroup (table)

The followings are objects from the new database by the former DBA B:

COMMON(database)

OfficeMap (table)

Office (table)

OfficeType (table)

Person (table)

Role (table)

PersonRole (table)

PersonMenuGroup (table)

XML web service by developer A

The web service project was identified as:

Conservation.Organization.WebServices (.net project)

OfficeData.asmx (class)

PersonData.asmx (class)

Conservation Components (DLLs) by developer A

The component library projects were identified as:

Conservation.Organization.Information (.net project)

OfficeInformation.cs (class)

PersonInformation.cs (class)

Conservation.Organization.Data (.net project)

OfficeDataAccess.cs (class)

PersonDataAccess.cs (class)

Interview

One interview was conducted with the former DBA, who was behind the implementation of the renewed centralized database. The summary of interview was attached at the end in section C.4.

Discussions

The researcher has also participated in a few meetings when the XML web service and new .Net components were discussed and demonstrated.

System Testing

The research has tested the office web service for acquiring its functional and performance information.

C.2 Summary of Finding

Project Summary

The Office and Person data projects were aimed for addressing on-going issues arising from the lack of a centrally managed repository for core data about the agency offices and staff. For many business systems including Intranet, BioWeb, VAMS, Permissions, Land Register, office and person data are part of the application data set. They are important attributes for many data entities; they can be used for categorizing and reporting on data; they can be used for security control for the systems. For many applications, they have their own set of these data within their own application databases. Many attempts have been made since 2001 for setting up a central repository, but there isn't much success.

Early Attempt

A few documents were found from the very early stage around 2000 on the topic. For example, there was a very early system called Basic Systems for Conservation Managers (BSFCM), the requirements for person and organization (PERORG) were defined as followings:

PERORG consists of the database tables containing the common person information needed by BSFCM processes and applications. This information is not usually maintained by BSFCM applications, but in some cases BSFCM applications could be the owner of specific items of person data.

BSFCM requires person information for several reasons:

- Access control is based on the person's position and cost centre.
- Ownership of workplans.
- Assigning people to workplans and costing their time.

- Recording the indirect staff costs, including unallocated time.
- Specifying resources for detailed project planning and time recording.
- Interfacing to leave and payroll records.
- There is a potential future need for access to staff capability information.

The general areas of information required would be:

- Name and other general details.
- Time bounded cost centre information (host cost centre, access rights to other cost centres).
- Permitted level of access to Business Planning.
- Time bounded salary information, including leave and allowances.

There were some functional specification and software design documents for the PERORG system. Such as the following design extracted from the “PerOrg inetOrgPerson Object Class” document:

“inetOrgPerson object Class

The inetOrgPerson object class is a general purpose object class that holds attributes about people.

The attributes it holds were chosen to accommodate information requirements found in typical Internet and Intranet directory service deployments.”

The following is an example of the prototype screens extracted from BioWeb PerOrg Prototype Screen Dumps

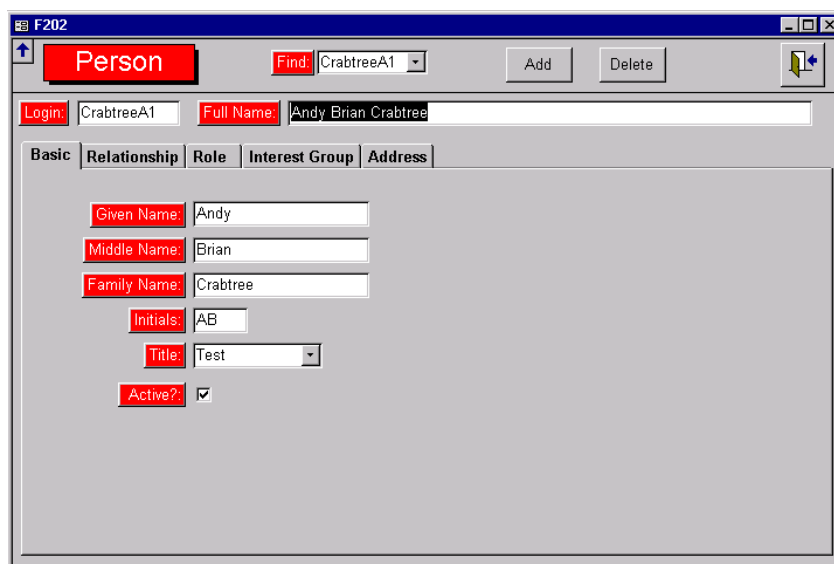


Figure C. 1 PerOrg Prototype Screen 1

The following is a system architecture diagrams for PERORG extracted from PerOrg User Requirements

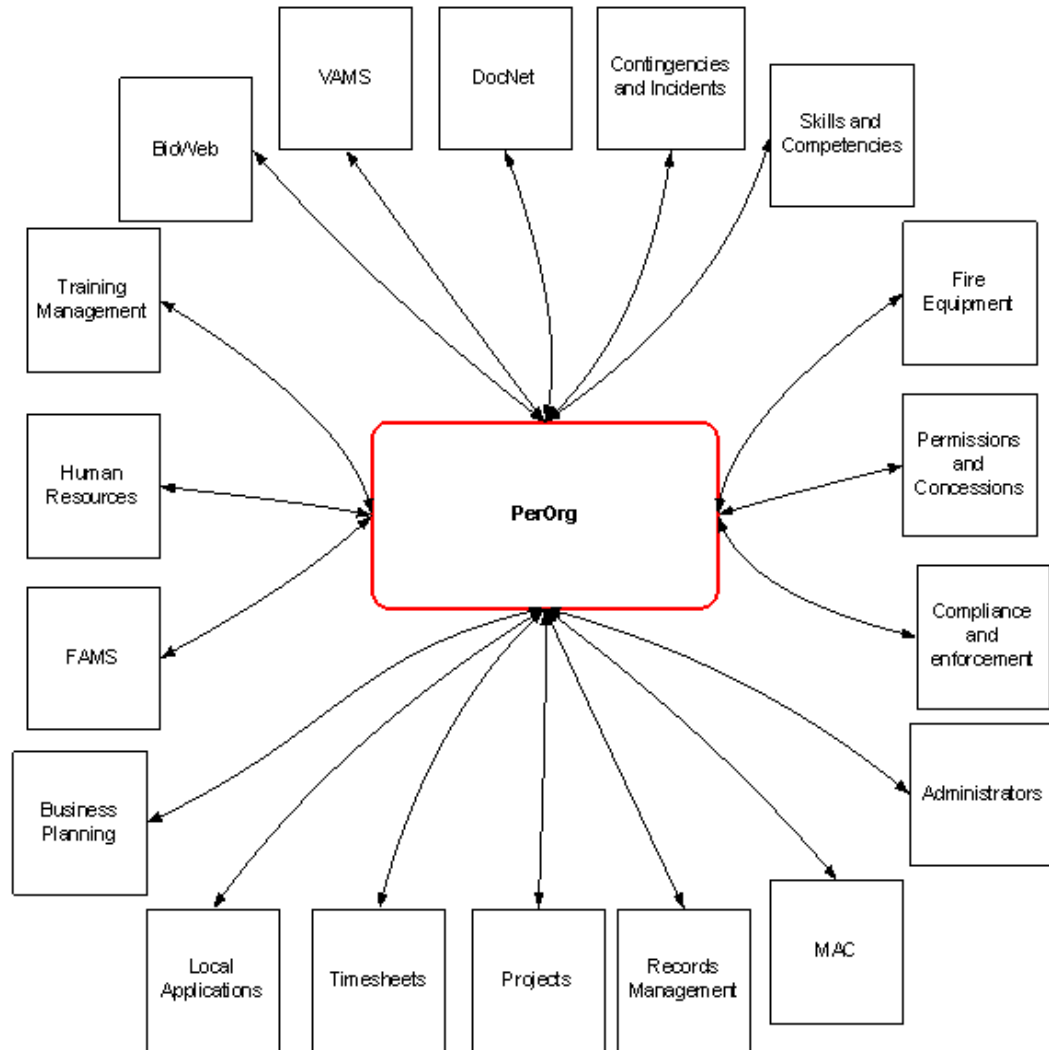


Figure C. 2 PerOrg System Architecture

From the documents and discussions, there is nothing found that indicates such system (PERORG) has ever been eventually developed and deployed to production. All early applications used their own set of office and person data, including the Intranet, Bioweb, Vams and Business Planning.

Early Integration

In 2002, when a new system (Permissions) was in development, the lead developer at the time was very keen to use a data sharing approach to tackle the issue. She promoted to use a common set of office and person data shared by other applications. The table Office in an existing database “DOC” was used for creating a view in the

Permissions database. The DOC database was originally created for the department's Intranet application. In later that year, when another new system Land Register was developed, another database CommonDB was created to hold extra office data such as the mapping of the offices IDs from different existing system, such as Intranet and VAMS. It was intended that the CommonDB would eventually take up the role as central repository for all office and person data for all systems. Since then when new systems were developed, it was always recommended that office and person data should be stored and retrieved from the central databases, but no progress has been made to merge the two databases, DOC and CommonDB.

The following was a diagram extracted from the discussion document "PerOrg Presentation". The document proposed the new common database:

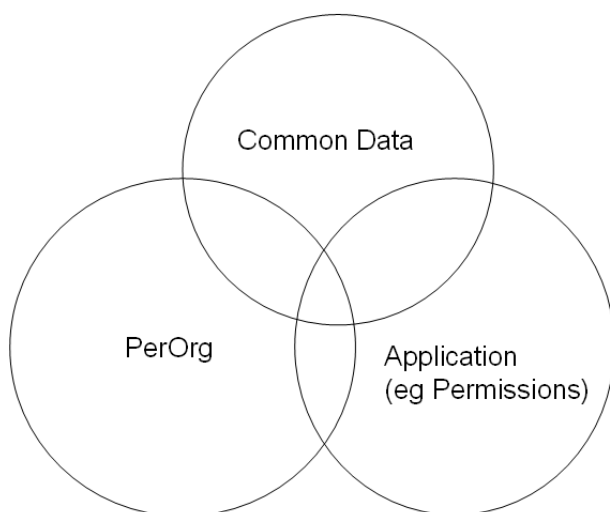


Figure C. 3 PerOrg with Common Data Proposal

XML Web Service

In 2006, an XML web service was created for providing office data through the web service interface. It was originally developed for the Bioweb application. The requirements for Bioweb for office and person data are relatively simple. They are mainly for read only access and populating drop down lists in the application screens. Although the XML web service was not designed for requirements of other systems, it was promoted to become the solution for office integration. Issues and problem arose when other systems such as Permissions tried to use the web service functions, as most of the stored procedures in the Permissions database need some sort of office and person data, but it is very difficult for those stored procedures to communicate with the

XML web service to retrieve the data. The conclusions were that the XML web service could not be used by Permissions in its current form.

The followings are the functions for the XML web service:

- [GetMultipleOfficeDetails](#)
Return the details for multiple offices. The 'codeList' is a list collection of string office code
- [GetOfficeDetails](#)
Return the details for a office
- [GetPickList](#)
Return a picklist (Value, Text) of offices
- [GetPickListOfAreaOffices](#)
Return a picklist (Value, Text) of area offices
- [GetPickListOfConservancies](#)
Return a picklist (Value, Text) of conservancies
- [ListAll](#)
Return a list of all offices
- [ListAllXml](#)
Return an XML document of all offices
- [ListAreaOffices](#)
Return a list of all area offices
- [ListAreaOfficesForConservancy](#)
Return a list of all area offices for a specified conservancy
- [ListConservancies](#)
Return a list of all conservancies
- [ListMajorXml](#)
Return an XML Document of major offices

Renewed Database Integration

When the department implemented a new SAP application to replacing the legacy VAMS system, the SAP implementation uses its own set of office and person data, which has created more complexity to the office integration for other systems. In a new attempt to tackling the integration, a new Common database has been created in 2008. All tables from the existing DOC and CommonDB have been copied to the new database.

The following is an entity relationship diagram re-created from the database:

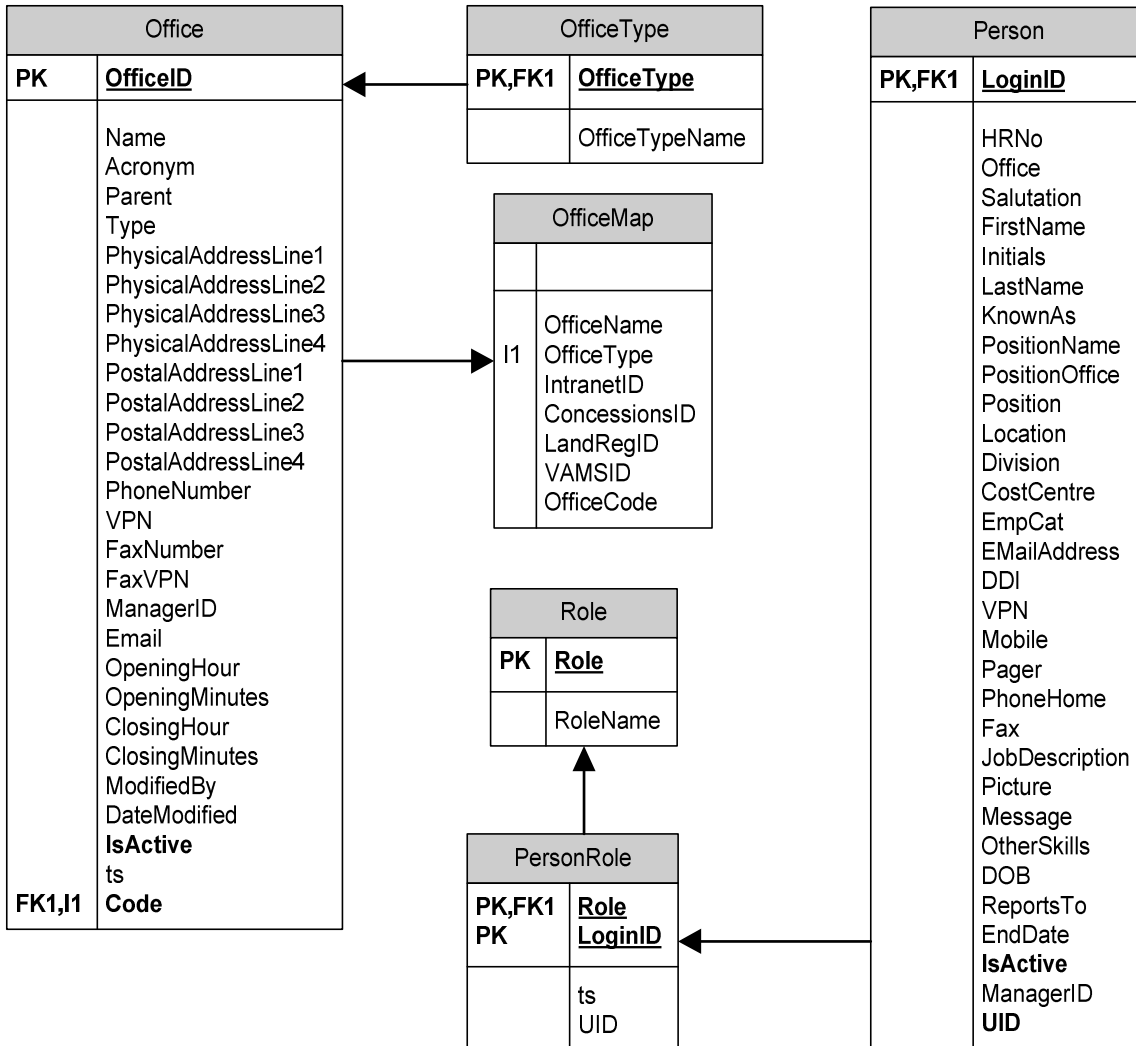


Figure C. 4 Office & Person Common Database ERD

Renewed Office & Person Component (DLLs)

Recently in 2009, the developer who has developed the XML web service has delivered another set of software components in this area. They are .Net class library components. The main assembly (DLL) is called Conservation.Organization.Information.DLL, the followings are the classes and their members:

```
OfficeInformation.GetDetails(String)
OfficeInformation.GetPickList
OfficeInformation.GetPickList(Conservation.Organization.OfficeType)
OfficeInformation.GetMultipleDetails(String[])
OfficeInformation.GetPickListOfAreaOffices
OfficeInformation.GetPickListOfConservancies
OfficeInformation.ListAll
OfficeInformation.ListAllXml
OfficeInformation.ListAreaOffices
OfficeInformation.ListAreaOfficesForConservancy(String)
OfficeInformation.ListConservancies
OfficeInformation.ListMajorXml
PersonInformation.GetDetails(String)
PersonInformation.GetPickList
PersonInformation.GetPickList(String)
PersonInformation.GetMultipleDetails(String[])
PersonInformation.ListAll
PersonInformation.ListForName(String)
PersonInformation.ListForOffice(String)
```

C.3 Preliminary Analysis

Case Study Questions:

- 1) *What is the Office integration project from different prospective?*

From the business point of view:

From many of the documents found in the research, especially from the recent 'Office and Person' document, it said "It has long been recognised that the agency has a number of applications that are very dependant on accurate information about

office and staff (and the relationship between the two) while at the same time having no enterprise-level strategy for managing this information and no central register of this type of data.” The main objective of all these projects was to establish a central repository for office and person data for all business system in the agency.

From the documents, there have been a few initiatives from the business identified. From a very early discussion paper, the project (called PerOrg at the time) was to simplify the system maintenance work because “There is a need for ongoing manual maintenance to keep the staff information up to date”. In later approaches, they included establishing and sharing a central database, using standardized methods and procedures (such as XML web service and common DLLs) for office and person data. Overall, for business, as it was summarized in a more recent document, the project was “to maintain consistent and secured office and person data for all the department’s systems while making the use of them easy.”

From the system development’s point of view:

There were a few system designs and some deployed components for the office and person projects, they are summarized as following:

- i. To develop a standalone system dedicated for maintaining office and person data, as it was described in the early design document.
- ii. To establish common and shared database tables, as it was started in 2002 and when some of the existing tables were originally created.
- iii. To develop software components (XML web service, DLL) for the usage of office and person data, as it was shown in many of the recent documents and component projects in source code.

2) *What are the goals for the project?*

From business point of view

For different approaches in different stages, there were different business goals identified:

- i. When a single system like PerOrg was proposed, it was to produce a single system that will service all the other applications, as it was said in the project document.

- ii. When the shared tables were proposed, they are for used by different systems at the same time in the ways individual system may require. Because the share tables are in separated database from other applications, there are less constrains on the business applications.
- iii. When the XML web service and DLL component were developed, they were primarily for sharing among applications in the system level and easiness of their use, as argued by the developer during the software demonstrations.

The business goals for all those projects were specified in the recent discussion document; they were for maintaining or improving the accuracy, security and consistency of the office and person data and their usage for all business systems.

From system development point of view

For different approaches in different stages, there were different development goals identified:

- i. When a single system like PerOrg was proposed, it was to develop a centralized user interface to manage office and person data for all in-house systems.
- ii. When the shared tables were proposed, it was to develop a centralized database used by all in-house systems.
- iii. When the XML web service and DLL component were developed, it was to develop a centralized application programming interface to the office and person data.

In summary, they were of enabling effective sharing of the office and person data among all in-house business systems.

3) *Why is it important?*

As specified in the early PerOrg project, there were risks to the business if the project fails

- i. “Redundancy of data resulting in multiplication of maintenance effort.
- ii. Multiplication of development effort in developing duplicate sub-systems.
- iii. Inaccurate, inadequate, or unavailable data resulting in inability to make timely contact resulting in loss of property or life.”

For the common database approach, they were important because:

- i. Without duplicated office and person data in multiple databases, they would more likely be consistent and accurate.
- ii. A common database independent of a single business system would give flexibility to development for all business system.

For the common software component approach, they were important because:

- i. They were easy to use and would reduce development cost for other system.
- ii. They would provide a standardized user interface for all systems and improve system usability across the department.
- iii. They would become part of the best practice for system development for the agency.

In summary, the projects were important for data integrity, security and system development. If the integration have succeeded,

- i. We would have eliminated duplicated office person data and the cost for maintaining them.
- ii. We could ensure all business systems have consistent and up-to-date office and person data, and no confusion for end users.
- iii. We could re-use components for displaying and handling office and person data among business systems, while reducing cost.

4) *What are the solutions that have been used so far?*

By searching through the documents and source code, it was found:

- i. The PerOrg system has never been developed, although there were some system design documents indicating that it would be a client-server application.
- ii. The common databases approach was implemented in Microsoft SQL server database, as for all other business applications.
- iii. The office & person XML web service and DLL component are all developed in .Net C#.

5) *What are the critical success factors for the integration project?*

For the PerOrg project, it was argued in the project proposal document that critical success factors for the project were:

- i. “The system is made available to other projects with the application program interface (API) defined. It shall have been used on at least one application, and made available to other developers via the intranet.
- ii. Stewardship and enabling processes have been agreed and implemented for at least one application domain.”

In fact, as the researcher found that project development was never started, most likely because it was not funded. It would be concluded getting management’s support and funding would be one of the most critical factors.

For the common database approach, it was argued by the Permissions lead developer in the discussion document that critical success factors for the project were:

- i. “The new system (Permissions) was integrated with the existing system (VAMS) using the tables in the common database.
- ii. Other systems would be changed to use the common database.”

For the XML web service and component approach, it was argued by the Bioweb developer in the team discussion that critical success factors for the project were:

- i. The new web service and component would satisfy the requirement from many business systems.
- ii. They are easy to use and adaptable for different systems.

In summary, critical success factors for the project would be:

- i. Clear and basic business requirements – They are important for getting management’s support. For project like PerOrg, its requirements were complicate but they were only something used by existing systems, it was difficult to convince the management.
- ii. Management’s support – Because there would be no proper funding without it, like PerOrg, although there were a few attempts as identified in the research, it has never been resourced and development was never started.
- iii. Suitable software platform – They must be workable. For the common database, it was built in MS SQL server, same as for all other business

systems; it had made it possible and easy to implement data sharing. For the XML web service and component DLLs, they were developed in using .Net which made the development easier than in any other forms.

iv. **Business Involvement**

In all the attempts for office & person data integration, only people from one or two business systems were involved at one time, which caused problems of defining the requirements. In the PerOrg project, it was mainly aimed for Bioweb. In the database approach, it was for Permissions, and in the web service and .Net component, it was developed by the Bioweb developer. The lack of involvement of other business systems at the same time was the main reason the requirements were not completed and the systems were not successful.

6) *What are issues and challenges faced by the project team?*

For the PerOrg project:

- i. It was a long running project from a long time ago as found in the research. Documents were created from 1999 to 2004. People involved were mostly business analysts, including one developer. The project did not have a set of firm requirements and system design in its entire life.
- ii. The project has not been funded, most work was done on discussion base, and no source code was identified in the repository after the researcher had done a thorough search.

For the common database approach:

- i. Although the Permissions development team made a decision to use office data from other database rather than its own one, very few other systems followed. The research has found only one another system developed after Permissions uses the same common database tables, no other existing systems have changed and most of them still have their own office and person tables.
- ii. The management issues for changing data in the common database could become very complex, as it was demonstrated in the recent organization change. When an existing office was disestablished and another one changed its name, it had cascaded impact to all the business systems all in a sudden.

For the XML web service and component approach:

- i. There are various requirements for different applications. While some are simple drop down lists, such as for the Bioweb application where the component was originally written for, but some are more complex. In Permissions system, office and person data are used for access control and are referred in many tables.
- ii. The components are difficult to use from within SQL server stored procedures. Components are designed for easy to use in the .Net platform, as they were demonstrated in the discussion.

In summary, the issues and challenges are:

- i. The requirements of the office and person data vary for different systems and they would be difficult to define.
As the DBA said, “Scope (its control and expending) is the biggest issue. There are too many systems involved. They all have different business owners, development techniques and business agendas. The coordination is almost impossible.” As consequence, it was difficult to get full support from management and funding, such as for the PerOrg project. The requirement for PerOrg was aimed for a small business system while it already had such function built-in. PerOrg was a standalone system and had no basic and firm set of requirements for a particular business process. The research found many document and design from many people involved, however most of them ended up un-used.
- ii. It was difficult to convince business to change existing systems without obvious benefits that can be seen from the user interface.
In the approach of using common database, no existing system had been changed to use it as the research had found. Part of the reasons would be the risk and complexities involved; most business systems like to have their own databases as the research found. The complexities of data change in the common database was a major factor too, as the research found there was never clear ownership on the office person data, and people would not be happy if something changed without they being notified first. Such change would also take long time because of the consideration of the impact to the

systems. The impact would include business processes, such as workload and revenue distribution.

- iii. It was difficult to develop a simple solution to satisfy various business needs.

This was the case for XML web service and .net component project. The XML web service and the component were simple project developed for the easy use for the Bioweb system, which mainly requires office and person data in simply forms, such as dropdown lists. The component had been used by one other system with similar need, but it will be difficult for other systems like Permissions to use, as XML and .Net component are difficult to be called from within the SQL server stored procedures. The main concern was performance, as office and person data are needed in most of Permissions procedures for basic business rules including security access.

7) *What are the major technical issues and solutions for the projects?*

For the PerOrg project:

- i. The requirement for the system had not been clarified and supported by business. Although there were a few attempts on trying to define the system requirements as recorded in a number of documents, they were done by different people in different period, and they were not consistent and none of them had clear support from the business or management. Otherwise it would have been funded and put into development.
- ii. The system design had not been clarified. As the result of no project funding and confirmed business requirement, although an entity relationship diagram was drawn in a proposed design document, no other system design had been found, such as system architecture. From message records, it was indicated the proposal was done as time filler by the analyst who happen to have some spare time.

For the common database approach:

- i. To establish common database.
This was achieved by the Permissions lead developer at the time in 2003. She convinced the DBA and the manager at the time and got the new CommonDB database created. From the documents, she seemed very

proactive and persistent. She had developed the Permissions system in a short timeframe.

- ii. To integrate with other business systems.
The Permissions lead developer had tried hard pushing for integration of other business systems to use the commonDB as the research found in many document and discussion records. But no obvious result has been found, except one later and small application, no existing systems have been changed to use CommonDB. The latest SAP (ERP) was implemented with its own office table, and keys for the records has to be matched using a table in a separate integration database called 'MiddleEarth' as described in an integration document.

For the XML web service and component approach:

- i. To develop the component for the use of single system.
It was achieved by the Bioweb developer within the Bioweb project. The developer had explained and demonstrated the web service and the DLL component in a few occasions recorded in discussions and documents.
- ii. To develop the component for the use of other systems.
The Bioweb developer claimed the component was open for any change and improvement in discussions. According to the developer, he was currently working with another developer on the improvement of the component. The researcher had not found any major changes of the software from what the Bioweb developer had originally published.

In summary, the major issues and solution for the office and person integration would be:

- i. Missing of clear and basic business requirements
This would be the biggest issues for the PerOrg project. Many requirement and proposal documents had been found for the project, but they have different content and were created by different people. Most of them were aimed for development of a standalone system, which would be complex but without clear targeting systems. Because of the lack of support from the management, systems had not been developed, and no solution would be identified.

- ii. There would be too many system involved with different technical requirement.

As illustrated in the PerOrg architecture diagram, there were too many in-house systems, although they would not be necessary for all of them to be included. The XML web service and .net component could not be used by systems like Permissions for good technical reasons explained in the findings. It would be very difficult to develop a ‘silver bullet’ if it is not totally impossible.

- iii. Missing of suitable software platform

Much effort had been spent on finding or define the software platform for the integration. According to the documents, The PerOrg project was designed as client server system with API interfaces. The common database was built in MS SQL server, and the XML web service and component DLLs were in .Net. The research had found that SQL server and XML web service would be shared by most of the business systems.

8) *How have XML web service been used and to what degree?*

As described in the findings section, XML web services have been developed in some stages of the project.

- i. They were developed by a developer for a particular business system (Bioweb) originally. They would be used by other business systems according the developer, as most services were generic functions.
- ii. The service provides a basic set for functions for retrieving office and person data. As in their description, they are for simple purposes like populating dropdown lists. No updating and linking functions were identified.
- iii. The XML web services were replaced by a newer set of DLL components later from the same developer. The source code was investigated during the research.

9) *What are the reasons for not adopting XML in the projects?*

- i. No argument was found in those PerOrg documents why XML was not used. The project was designed as traditional client-server application.
- ii. The Permissions developer argued in one of her documents, having shared common database for office and person was the simplest and easiest way for

the integration. The common database has been in use since. The research had found no major issues with it.

- iii. According to the Bioweb developer, the DLL components were much easy to use and would hide all the complexity of the XML web service. The research had found no performance difference for the use of DLL component or XML web services.

10) What are the reasons for adopting XML in the projects?

The main reason for developing the XML web services, as argued by the developer, was that they would be shared by other system, as many of the business systems already used XML web services to access backend databases and other source.

11) How were the decisions of the system design made?

The research found many discussions for many years for trying to reach common agreements on the project.

- i. Business analyst made the decisions.

For the PerOrg project, the business analysts had outlined some basic design in the requirement and proposal documents.

- ii. Developers would also push for their solutions.

The Permission developer was the main driver of the common database approach; she was also the author of a few documents and presentation. She managed to get agreement from the manager and DBA at the time.

Permissions system was the first to use the common database. This also happened in the later approaches on the topic, when the Bioweb developer designed and developed the XML web service and DLL component. In discussions, he also quoted he got support from the senior developer and he wanted support from peer developers.

12) How successful were XML web service integration solutions when XML was actually used?

The XML web service was not very successful, because

- i. The research found that it had not been used by any other systems except Bioweb that it was originally designed for.

- ii. It is now replaced by DLL component developed by the same developer. The developer argued DLL components were much easier to use.

13) What are the critical success factors for the XML application(s)?

From the investigation of documents and discussion with the developer, some factors were identified

- i. There should be comprehensive design objectives.
There is a lack of design objective for general integration. According to the developer, he designed the XML web service for look up data used in the Bioweb system, although he promoted it to be used by other systems, he had no discussion with developers or business analyst from other systems, neither had he worked in other business systems.
- ii. Advanced XML Technique should be used if necessary.
Only basic XML web service functions were provide as shown in the findings, no advanced XML technologies, such as XQuery had not been found. The functions were for simple requirements like populating dropdown lists, which would not be used as source for dynamic queries or other data manipulations such as data joining.

14) What are the reasons for the XML web service integration solution being not successful?

The reasons for the office XML web service being not successful were:

- i. The XML web service was slower than direct database access.
The XML web service is much slow than direct access to the database. Comparing to using view to access office data in the Permissions database, the researcher found it would not even possible to use the XML web service to get the same data for the large number of stored procedures.
- ii. The XML interface functions were too simple.
The XML web service only provides basic data through a set of pre-defined functions. However, business systems like Permissions use the office as basic data set and these pre-defined functions were far less than enough. Adding or modifying those functions was less flexible and dynamic than working directly on the views within the database. As the DBA said, “XML is not suitable for publishing sensitive or secured data.”

- iii. XML data was lack of database referential integrity feature.
- The XML web service could not be used to enforce data referential integrity in SQL databases, as there is no direct links between XML web service and the database. Separated checking processes needed to be developed and implemented. However, using stored procedures and triggers would achieve that if the office data were from views in the database, which would be much easier and simpler than those separated processes. As the DBA said, “XML is not as good as SQL tables, such as for querying and joining.”

C.4 Summary of the interview with the former DBA

- 1) What are the business goals for office & person data sharing project?
- To create and maintain a single point of entry for the common data, the benefits lay in
- Consistency and credibility of the data
 - Maintaining a good control and ownership of the data
 - Providing a high availability of the data
- 2) What are the system (technical) challenges for the integration project?
- Challenges are from the following areas
- Data Matching – the current multiple data sources have mismatching data.
 - Data Ownership – There are no clear owners for the current data, neither for the proposed common data
 - Data Security/compatibility/performance issues
 - Creating and maintaining the single database
- 3) What is the biggest challenge for the project?
- Scope (its control and expanding) is the biggest issue. There are too many systems involved. They all have different business owners, development techniques and business agendas. The coordination is almost impossible.
- 4) How can XML web service be used to address these challenges?
- The web service is a good solution, especially for providing a reading access for all possible users, but it may not be able to enforce security control on the data.

5) What are the solutions and how do they work?

The current solution maintains two interfaces. One is the direct access to the common database; the other is the XML web service interface.

6) Was XML involved in these solutions? And how?

XML web service provides a read only interface to the common data. It is of text format, so it can be used by any system. It also hides the schema of the backend database.

7) What are the reasons for adopting or not adopting XML in system integration projects?

XML web service advantages:

Help to provide a single point of data access for everyone.

XML web service disadvantages:

Not suitable for publishing sensitive or secured data

Not as good as SQL tables, such as for querying and joining

8) How successful were XML web service integration solutions when XML was actually used?

The web service has been used by one of the existing systems, i.e. Intranet.

9) What are the reasons for XML web service integration solutions being successful/not successful?

It is a success. The XML web service is quick and easy to use, it is always available. The data output can be used by any system for any process.

10) What will be the future plan and solution for the project?

To maintain the two interfaces including direct database access and XML web service.

Extra Interview Questions:

- How do you see (describe) your own role in the office data integration project?

A DBA, responsible for creating and maintaining the database, to ensure developers and business applications can have adequate access to it. I have also tried to convince others to use the common database, not their own set of private data.

- What other comments you want to the make on this project or other projects?
The common database must have a single and responsible owner.
XML web service can be expanded to provide data to users from other government department.
The corporate directory has it own database, it uses triggers to update Active Directory, MS Exchange, and Intranet. The setup is too complicate and hard to test.

Appendix D: Case 4 Data Collection & Preliminary Analysis

D.1 Data Sources

Documentation

Documents are primary source for this case. Corporate Directory (CD) project was outsourced to an external software house (referred as the software company) for development initially, it was then largely taken back by the department after the mid project review. There are a full set of the documents for the project, including the business case, user requirements, system architecture and design, functional specifications and a full set of source code.

The followings are the list of important documents reviewed for this research.

- Corporate Directory Final Report (by an external IT consultancy contracted for the review)

- Corporate Directory Design Document (including many versions, by the software company contracted for the development)

- Corporate Directory - Web Forms Answers (discussion record between the software company and the former CIO)

- Corporate Directory Business Rules (by the software company)

- Corporate Directory AD-Exchange As Built v1.5 (by the software company)

- Corporate Directory Pseudo Code (by the software company)

- Corporate Directory Business Case (by the former CIO)

- Corporate Directory - Post project review (discussion record by the system analyst)

- Issue register (by the project co-ordinator)

- JADE Web Service Processing (discussion and agreement between the HRIS vendor and the agency)

- Corporate Directory View Identity Test Matrix (by the test analyst)

- Corporate Directory Reporting Line UAT (by the test analyst)

- Concern for Corporate Directory (by developer A as his departing notes)

Source Code

The agency had hired two contract developers one after the other to take over the user interface development using ASP.Net. Most of the work was done by the developer hired in later stage.

Interview

One interview was conducted with the former system analyst. Summary of the interview is attached at the end in section D.4.

D.2 Summary of Finding

Project Summary

The Corporate Directory project began formally in May 2006 when the business case was approved, with a 6-digit amount of budget approved. An external software company was selected to complete the software development, it was started on 1 July 2006 and due for completion on 30 August 2006. According to the plan, around half of the project budget was allocated for the payment for the work done by the software company. In February 2007, when the project was in six months delay than the original plan, a project health check was conducted. The software company claimed that the project was 95% complete. The review recommended for continuing the project. It also recommended clarifying the project scope; preparing a new schedule; dedicating new resources to it; appointing a departmental project manager and ensuring that the steering committee and user group were actively involved. A technical working group was also set up. An extra amount of money equal to the original budget was allocated to the project and for the software company. At that stage, the project management office estimated that the project would take another six months to complete, i.e. which would be by August 2007. However, the project was still unfinished when the new CIO decided to suspend it in June 2008.

Original Plan

The following is the original plan for the project extracted from the business case document:

TASK	TIMING
1. Prepare tender document for Corporate Directory application	1-16 June 2006
2. Receive proposals, evaluate, finalise selection	19-23 June 2006
3. Map information flows to Corporate Directory	1-16 June 2006
4. Identify and define business rules for information exchange with other systems	19-23 June 2006
5. Examine existing key identity data for inconsistencies and resolve (using internal resources)	1-30 June 2006
6. Purchase software & hardware & establish test environment	26 June – 7 July 2006
7. Develop HRMIS database triggers	10-28 July 2006
8. Pilot test of solution	31 July-4 August 2006
9. Implement full solution	7-25 August 2006

System Design and Development

There were a number of versions of the system design documents identified during the research. According to the design document from the software company, the Corporate Directory solution focuses on consolidating people information from different sources, simplifying and automating data flows, allowing administrators to better control the environment, and provide the business with accurate, consistent person information.

The following table extracted from the “Corporate Directory Design” document illustrates sub systems which were included in the initial deployment scope.

Solution	Meets Percentage of Goal	Implementation Rating	Importance to Organization	Selected Priority	Implementation Order
Integrate HR, employee data, and intranet person profile	80%	Within ability	High	High	First
Integrate HR employee data and Active Directory and MS exchange provisioning	80%	Within ability	High	High	Second
Provision telephone data sources	25%	Within ability	Medium	Medium	Fourth
Web forms to enable the single entry point for raw data for HR		Within ability but may expand scope	High	High	Parallel stream developed
Distribution list population	100%	Within ability	High	High	third

Table 1 from Corporate Directory Design Document

The system data flow model extracted from design document shown below illustrates the data flows within the system. The interface for adding or modifying information is via a web form, this allows for the greatest flexibility and distribution as no client software set-up is necessary, and paper based re-entry can be supported via a printout option on the form itself.

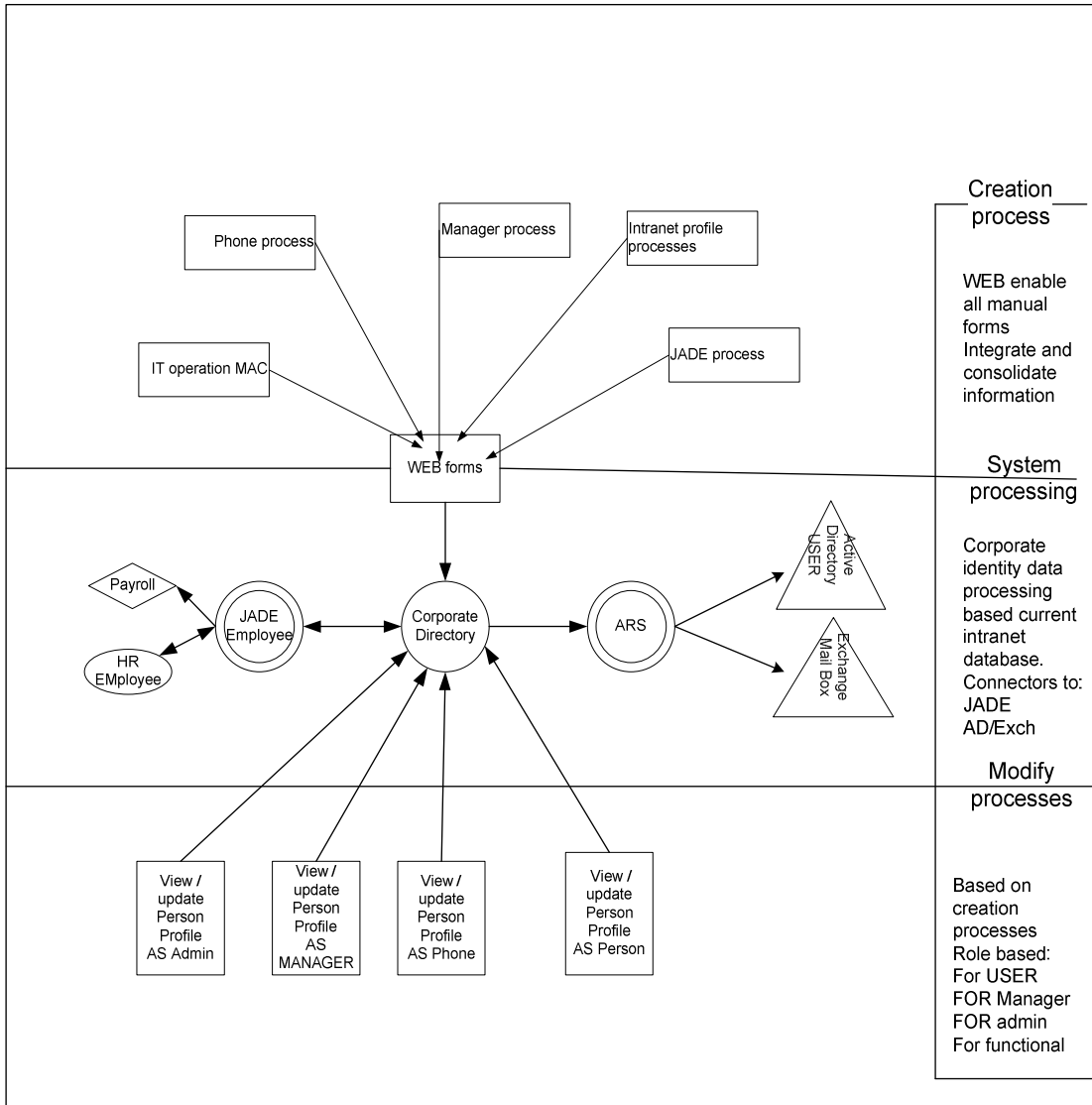


Figure D. 1 Corporate Directory Data Flow Diagram

According to the source code, the main pages for the wed form based application include:

- Default Start up page
- Appoint New Position

- Create Identity
- Modify Name
- Modify Work Details
- Outstanding Approvals
- Outstanding Manager Approvals
- Personal Details
- Reinstate Identity
- Request Unpaid Leave
- Terminate
- View Identity

XML Web service

In later stage, a large number of web services were added to the system design as interfaces to the HRIS (Jade) system. Some of these are for providing the Jade reference tables data that are used within the web forms, as lookup tables. The following Jade Web Services are used to extract the current data for the given Jade reference table used within the various CD web forms as lookup tables:

Jade Web Service Name	Corporate Table Name
GetEmployeeStatusList	Jade_Emp_Cats_Code
getEmployeeTypeList	Jade_Emp_Type_Code
getEthnicityList	Jade_Ethnicity_Code
getTaxCodeList	Jade_Tax_Code
getTerminationReasonList	Jade_Termination_Reasons
getBranchList	Jade_Conservancy_Code
getCostCentreList	Jade_Cost_Centre_Code
getDepartmentList	Jade_HOCOAO_Code
getDisabilityList	Jade_Disability_Code
getPreviousEmploymentStatus	Jade_Prev_Emp_Status
getRecruitmentSourceList	Jade_Recruitment_Code
getRegionList	Jade_Regions
getSAPCostCentreList	Jade_SAP_CC
	Jade_Position_Lists
getContractTypeList	Jade_Contract_Type_Code

Table 2 from Corporate Directory Design Document

The following Jade Web Services have also been used for the CD integration with Jade:

Jade Web Service Name	CD Integrated Process
createEmployee	Create Jade
modifyName	Modify Name
modifyPersonalDetails	Personal Details
modifyPhysicalLocation	Change Physical Location
terminateEmployee	Deprovision Identity
modifyEmployeePosition	Modify Current Work Details

Table 3 from Corporate Directory Design Document

By going through the source code, it was found that the above XML web services were not completed; neither the web forms have not been completed to use the web service.

D.3 Preliminary Analysis

Case Study Questions:

1) What is the corporate directory project from different perspective?

From the business point of view:

It was found in the business case that “the project was to implement a simple application that will use electronic (online) forms to accept information about people starting with the agency. That information can then be fed into all processes required to set that person up. These particularly include payroll, IT and phone operators in the first instance, but would provide easy ability to expand to notifying others desired. The application would use standard elements of existing systems (eg Jade HRMIS, Microsoft Active Directory) to store validated information and pass that between applications as required, to ensure a consistent set of data per person. This information would also be used by the Person Search facility on the Intranet. Change requests (eg for a new software application) and business owner management of email and distribution lists, and details of departing people would also be handled through this application.”

It would be concluded that the project was to centralise and automate a set of manual processes related to people and IT systems.

From the system development point of view:

As from the business case, the project would create its own set of user interfaces and integration with a number of sub-systems. It was to develop a new integration system to provide some new features to the business.

2) *What are the goals for the project?*

From the business point of view:

As high level business goals documented in the business case, they were as followings:

- i. “First, it is to provide a single interface to managing the information and processes that relate to a staff member in the agency, especially for starting and leaving.
- ii. Second, it will provide an important building block to simplifying IT logon processes in the future, especially for users of multiple applications that each currently require different user code and passwords. It will also in future allow us to easily manage access control for people outside the agency logging on to our systems, and our staff logging on to other peoples’ applications, in a secure and trusted fashion. As the system analyst said, ‘it would be the central point for maintaining security access for all business systems.’
- iii. Third, it will help the future integration of HRMIS.”

From the system development point of view:

As from the business case, the project has ambitious goals and involves a number of sub-systems for creating some new automatic processes. It would also fill in the gaps some other projects had tried but failed to do in the past, such as PerOrg.

3) *Why is it important?*

According to the business case, the current processes were inefficient. At present a number of different processes, forms and systems must be completed by business

unit managers when people join or depart the organization, which is time consuming and repetitive. In addition a manual form does not enforce provision of valid data, causing follow-up discussions to clarify requirements at times. This information is then stored in several different systems, requiring much duplication of effort. The information thus created is often inconsistent due to differences on the forms, in data entry or as a result of time delays for different parties to be advised/do data entry. As a result, the ability of staff to quickly discover reliable information on who a person is, what part of the agency they work for and how to contact them is compromised. When a person leaves, personal records tend to be slow to update except payroll, as a result there can be security issues with active IT user code related to long departed staff.

In a discussion with the business analyst involved in the project, he confirmed the above situation was true and he also added, a further inefficiency in such approach was that people who use multiple applications in the agency that each required a logon must remember multiple user code and passwords. This was very frustrating, and likely to cause staff to note these insecurely as they are too difficult to remember. The project would be a key building block to implementing Single Sign On, which enables all security access to our applications to be managed with a single logon per user. Such view has also been recorded in a discussion paper.

4) What are the solutions that have been used so far?

Core Components	Core Technologies
Corporate Directory Forms (User Interfaces)	ASP.Net Forms
CorporateDirectory database	Microsoft SQL server
DOC database	Microsoft SQL server
Jade Web Service (external, unfinished)	XML web service
MS Active Directory (external)	Proprietary
HRIS (Jade) (external)	Proprietary
Active Role Server (ARS) (external)	Proprietary

5) *What are the critical success factors for the integration project?*

There were two factors identified in the business case document:

- i. All involved parties are fully committed to the objective of the project and available to assist its implementation. It was initially achieved by getting all of them to sign off the business case.
- ii. Highly skilled resources are to be used. This was the main reason why a third party software company was selected to implement the system. The company was thought to be having the expertise and currently doing work for other government agencies.

Other factors were identified in the final review; they were also reflected in the issues register and other discussion documents. They were identified as the major factors for the failure of the project.

- i. Business requirements should be agreed and well documented. There was no initial requirement analysis of the system, as no time and budget was allocated in the business case. The business owner has only confirmed in a later stage of the project after the project health check.
- ii. Projects should be well managed and supported. There has been no proper project manager assigned to the project. Initially, it was supposed to be managed by the external software company, who have no access to the sub-systems or resource to handle them. Later on, a project coordinator was assigned but without actual accountability. The project team has no effective ways to deal with project issues.
- iii. There should be in-depth knowledge to the sub systems or support from people with such knowledge. The lack of knowledge of the HRIS jade was identified as a major factor hindering the development. It was also identified in the developer discussion that the reliance on the third party component to communicate to Active Directory was also proven as a problem, as it was very difficult to set up and not so reliable.
- iv. The System should be well design with proper interfaces defined. There were many changes to the system design as many version of the design document were identified. It was concluded in the final review report that the software company has no expertise in building web form interface in Microsoft technologies, which was raised as a major concern long time ago

by the agency developers. There was no clear interface design in the business case and the business requirement document.

In summary, the followings would be the most important critical success factors for this integration project.

i. Clear business requirements

The final review report identified that this was the most important factor, which was in line with the findings from the Standish Group (2004) report. It was also reflected in the discussion and other discussion document in the corporate directory project. In many situations, it was referred as changing scope or scope creep in the final report and other documents including the issue register.

ii. Project Management

The final report had identified it as the second most important factor for the failure of the project with many reasons. The project started without a proper project manager appointed. The project was allowed to keep going with extensive delay. In other documents including discussions and the issue register, the project team claimed they felt very difficult to report project issues as no proper management structure was set up to allow them to do so.

iii. Skilled resource and in-depth knowledge of the sub-systems

It was identified in many documents. The sub-systems in this project included Active Directory and external HRIS system, which were all commercial proprietary systems. Resource with in-depth knowledge of them was hard to find. Because of the lack of skill in these systems, no proper interfaces had been designed until the mid-project review. The XML web service was not implemented in the end because the project has run out of time as found in the final review.

iv. Business Involvement

The final report had identified that the lack of business involvement in the initial stage of the development had cause a lot of issues, including the requirements were unclear and kept changing. Although the business had involved a lot more since the mid project review and it did put the development back onto the right track, it came too late to save the project from failure.

6) *What are the issues and challenges faced by the project team?*

According to various sources including issue register, email records from all parties, and the final report. The major development issues were identified as followings:

- i. There was not proper project management.

This was identified as No.1 factor for the failure of the project in the final review report. This was also reflected in many other source including mid-project review and other discussion document. As the system analyst said, “The project was not set up properly from the very beginning. I was assigned to take it over after the crisis has already appeared.” The business case and initial plan had not clarified the project management. In some message records, it was identified that the CIO had assumed the software provider should have managed the project but they thought otherwise and didn’t do the job anyway. After the mid-project review, a new project coordinator was appointed, but no clear responsibility or authority assigned.

- ii. Business requirements were unclear and kept changing.

Since there was no initial analysis done when the business case was signed off and the project started, as pointed out in the final report, no formal requirements were documented initially for the project. There was also no active user involvement in the development before the mid-project review. Many of the email messages during the development were of discussions trying to clarify and agree on system requirements among the development team and various users group.

- iii. The system interfaces were not clearly defined.

As indicated in the initial plan and system design, the software company has largely underestimated the complexity of the issues from the sub-system, especial the HRIS system. Many issues had arisen during the development and recorded in the email records. Because of the lack of knowledge of the sub-systems, there was no clear interface design before the mid-project review. XML web service were only designed and agreed on later after the review.

- iv. The software company has limited expertise in HRIS and Active Directory sub-system.

There have been a lot of issues regarding the lack of knowledge of sub-systems. In the written record by a departing developer from the agency who had been involved, he was strongly critical of the software company for lacking security control in the system design and development around the HRIS system. In another document, another developer also question the use of a third party component ARS, as it was extremely difficult to set up or change. The component was used to communicate to Active Directory as the software company would not had other better way, which was proven to be problematic one.

- v. The software provider has limited expertise in development using latest MS technologies.

This was pointed out in the final report, and also reflected in email discussion records from developers involved in the project. The research has found that most the code developed by the software company were within the SQL server. The software company could not develop any user interface such as web form in ASP.Net, which is the department's standard platform.

7) *What are the major technical issues and solutions for the integration project?*

They were identified in the research as followings:

- i. Business requirements were unclear and kept changing.

After the mid-project review, according to the review recommendation, a new business analyst was assigned to the project in charge of the system requirements. Business users had been more involved and become aware of the consequence of unsettled requirement. All other possible effort had been made to ensure the requirements were clear and stable, including using the change request mechanism and the issue register.

- ii. The system interfaces were not clearly defined.

It was resolved in the later stage of the project by agreement with the HRIS software provider that custom designed XML web service were to developed and provided by the HRIS system provider. A few of the document for the interface design and contract agreement were found. However, these XML web service were not deploy or tested when the project was suspended. The researcher has identified temporary database that was used by the web form system, as there were no XML web service were available at the time.

iii. The software company has limited expertise in HRIS and Active Directory sub-system.

The situation had only been improved after the agency took back the web form development by using a contracted developer.

iv. The software company has limited expertise in development using latest MS technologies.

New experienced developer has been hired as contractor dedicated to the development of the system. A full set of ASP .Net project code were inspected in the source code repository.

8) *How XML web service been used and to what degree?*

As found in the final version of the system design document, the XML web service were designed for populating reference tables and carried out function call to the HRIS system. XML web services were adopted after the mid-project review, as recorded in the version of the design document. The initial design was to communicate with the HRIS system through SQL stored procedures; it was proven very difficult in the agency environment after a long delay in development. The HRIS system is a commercial system, no direct access to its tables were available, as it was confirm in the email record.

Even after the XML web services were agreed, there were complains from the developers in email records saying that the using of the XML web services kept running into errors, as the HRIS underlying database were often changed without notification and the XML web service were out of sync with those tables. As the project was running out of time to the end, the agency's developer had to set up a local database for the development purpose. The XML web service functions have not been used at the end, as the researcher had not found any source code to do so.

9) *What are the reasons for not adopting XML web service in the project initially?*

The final report found the software company lack of development knowledge using current technologies, such as ASP .Net. It would not adopt XML web service for the same reason. As from the initial requirement and system design document, interface to the sub-system were not clearly defined. The initial approach mainly consists of stored procedures and triggers in the SQL database. This was first told by the business analyst in the discussion, and then confirmed by looking into the

SQL database. The system was likely from the past experience of the software company, but it would not work in this case, especially for the HRIS system, as the underlying tables were not accessible. Although the software company spend a lot of effort up to the end of the original project plan and claimed 95% complete according to the review document, the interface of the initial design continued to cause issues and was eventually replaced.

10) What are the reasons for adopting XML web service in the project later?

According to the project plan and other records, there were some major changes in the system design after the mid-project review, because of issues of the requirements and system design problem identified in the review. XML web services were adopted, as they were the most feasible options available. In the email records, the HRIS provider recommended the XML web service and agreed to provide them as a service contract. The proposal was also well received by new the agency developer. The new ASP .net web form application would simply make use of those XML web service and the development tools were included in the agency development platform.

11) How were the decisions of the system design made?

It was mentioned in the business case that the software company was selected because it has done similar job for other government agencies, the initial design was most likely carried over from its previous work. Such thought was also expressed by the business analyst in the discussion with the researcher. The initial design was also the result of the lack o knowledge of latest MS technologies, such as ASP.net and XML web service, this was first reflected in a former developer's email record of his concern to the project. It was identified as one of the major issues in the final report. There were numerous issues with the initial system, the research has found over three hundreds issues were recorded by the test analyst in defect tracking system. The decision for the change of the system design was reached after the mid project review, and the project has used up the original budget at the time.

After the mid-project review, the development team and business user had become more involved. Clarified business requirements were agreed by business owners and the new development team. Technical decision such as adopting XML web

service was recommended by the HRIS provider as preferred option and agreed by the agency, as the initial design was proven non-workable and needed to be replaced.

12) How successful were integration solutions without XML web service?

The non-XML web service interface approach was not successful at all. It was identified in the mid-project review. In the defect tracking document, it was identified many issues was related to the non-XML interface to the HRIS system. In the developer email and final report, it was pointed out that the problem associated with setting up and maintaining those stored procedures and triggers which function as interface had always been problematic and cost excessive amount of time, it was one of the main factors for the failure of the project.

13) How successful were XML web service integration solutions?

As from the project plan, XML web service interface was adopted after the mid-project review. The HRIS provider was committed to build and deploy the web service, new contract was agreed and progress was made as indicated in the email records. The XML web service interface also enabled the new web form interface development to be done in the agency standard development platform, and improve the development of whole system. ASP .Net and XML web service skills are widely available, and the agency had many of the systems developed in such platform. The business analyst said in the discussion the hiring of the new contract developer was easy and his performance was outstanding. One of the factors would be XML web services are more widely used than SQL stored procedures and triggers, as the later ones require specialized SQL skills and the development tools are harder to use.

By comparing the work and progress of the project, regardless of the business requirement, the business analyst said if XML web service were used from the very beginning, the project would have a much better chance to succeed. As from the project plan, since the initial non-XML design had been used until half of the total budget had been spent, the counterfactual comparison from the business analyst would be reasonable.

14) What are the reasons for the initial non-XML integration solution being not successful?

As identified in various source, they were:

- i. There was lack of understanding of the complex integration nature of project. As in the business case, the project was supposed to be completed in three months. There was no allocated time and budget for initial analysis before the project was formal started. The project ended up far more complicate than it was thought. It was a major factor identified in the final report.
- ii. The initial design is unworkable, it was would be an assumption carried over by the software company. The business case mentioned no system design at all, but included critical successful factor that the selected software company has relevant experience.
- iii. The software company has no in-depth knowledge of the sub-systems. It was reflected in the system design. In the issue register, it was recorded that one of the critical software component ARS used for communicating with the Active Directory sub system was very difficult to set up to work properly. It would not be the case if the software company knew Active Directory or its tool well enough. The initial interface to the HRIS also encountered numerous problems and has to be changed after the review; it would not have anticipated that because it would have little knowledge of the proprietary HRIS system.
- iv. The software company only has limited MS expertise as it was pointed out in the final report. Because of this, it could not be able to adopt those latest technologies such as XML web service when problems were encountered in the early stage of the development. The new XML web service was only used by the new development team. After the mid project review, the agency has taken over the development of the web form user interface using ASP .Net and the mount of task for the software company was reduced as it was identified in the documents and source code.

15) What are the main reasons for the project being delayed and suspended at the end?

They were summarized in the final review report and reflected in many other documents and discussion between the business analyst and the researcher.

- i. The business requirements were misunderstood.

This was regarded as the No.1 reason. This view was reflected most source including the final report, issue register and discussion records. The business objectives in the business case document were ambitious but they were not reflected in the requirements properly. Business user only started actively involving with the project after the mid-project review. In the end, as by comparing the initial design and final project plan, the scope of project had been reduced largely; integrations with MS exchange and telephone system were taken out. It would be impossible to achieve all the initial goals included in the original plan.

- ii. The project was not managed properly.

According to the business case, the project would only need three month to complete. In fact, it had been delayed for a very long time and the development missed many milestones, but it was allowed to keep on going. The final report has identified many issues of the project management; it was found that it not even had a proper project manager from the beginning to the very end. It was described in a discussion that the project was thought a small 'pet' project in the beginning but had grown into an 'elephant'. As argued by Urban Nulden (1996), most IT project failures were management related, including being over-committed. It was truly reflected in this case.

- iii. The integration interfaces were not clearly defined.

They were not identified in the initial system design document. The initial approach of using SQL stored procedures was from the software company. These interfaces have caused numerous issues as recorded and were replaced by XML web service after the mid-project review. The initial interface design and the missing of clear interface was a major factor for the project failure, as it was pointed out in the final report.

- iv. The system development environment was never set up properly.

The final report has pointed this out as a major issue; it was also reflected in the complicate procedures for setting up the development, testing and

deployment environment. Because the sub-systems are not well understood or ready for integration, such as the MS Active Directory and the HRIS system. The installations of the sub-systems were a major task requiring a lot of expertise, which the software company was lack of, as identified in the final report. There were confidential data in the HRIS system, which would make it difficult to set up for development and testing if the environment is not well designed.

- v. The solution was poorly designed.

This was identified in the final report. There was no development method found in the project development life cycle. There was no proof-of-concept or prototype found during the research. The business case has indicated that it was supposed to be a simple project by the experienced software provider. In fact the project has entered into many unknown fields and encountered numerous issues, for example, the business analyst pointed out that no proper workflow component was used for managing processes and interacting with multiple sub-systems.

There are many other factors identified in the final report and other source for the failure of the project, most of them were of project management ones including the missing IT analyst in the beginning and the lack of issue reporting mechanism.

D.4 Summary of the interview with the former system analyst

The interview has only covered a limited set of the questions from the business perspective. No technical question was asked.

- 1) What was the integration project? (How and why was it started?)

It was an attempt to handle personnel data in one place for the organization and push them out to various systems including Active Directory, Intranet, and Human Resource Information System (HRIS).

- 2) Why was it important to organization?

- It was for improving productivity and filling data 'gaps' among various systems when staff leaving or changing.
- It was regarded as a quick-win for the CIO.

- The People Organization & Development (POD) wanted to use it to automate some business processes.
- 3) What were the solutions that have been used so far? How were they achieved?
- To create a central place for entering and maintaining staff information such as start date, position, salary and etc.
- To implement some workflow elements include generating email notification and steps for business processes related to employing new staff.
- 4) What were the major development issues and solutions for the integration project?
- There were no clear business requirements, they were not documented, and POD kept changing them.
 - The initial business representative from POD did not understand the business requirement and was made redundant at the end after 6 month in the project.
 - There is no analyst involved.
 - The original PHP user interface was forced to change to .Net bases web pages as per standard of the department, but no requirements or specifications have been passed to the agency's developer.
 - The software company managed the relationship of HRIS system vendor and the agency but did not understand the requirements of the agency.
 - There was no proper development environment set up for the project, as there were many systems involved and they were hard to set up, including the Active Directory, the HRIS and Microsoft Exchange. The test suite was used by the software company for development, which had caused further problems when the agency development team wanted to do testing and to get current version of the system.
 - There was no proper project management. The project was supposed to be managed by the CIO, then by the software company, then by the project manager of the agency.
- 5) What were the critical success factors for the integration project?
- Support from the business (POD in this case)
 - Project management
 - Business requirement

- Evaluation of the package solution
 - Robust system
- 6) How are those issues managed? What were the results and reasons of them?
- The support from the business was very weak initially as the initial representative from POD did not know the business requirements. After 6 months of the development, the project became critical, POD then started thinking about it more seriously and become more involved. It was the project manager who got the POD people to realize the problems.
 - There had not been dedicated project management initially as it was thought the project would be finished in less than 4 months. The project manager from the agency was brought in after the mid project review for trying to put the project back onto the right track. But it was too late by then and he could not save the project.
 - The requirements were never properly done by an analyst. The analyst was brought in later but since the project had already been in crisis, he could only manage existing issues and try to get the system delivered as soon as possible.
 - The initial solution was a faulty one and would not work for the agency. The software company had made changes to it since and the agency had taken back the development of the user interface.
 - The system was never robust and it had never reached the stage that was stable enough for deployment, although dedicated test analyst resource had been assigned to the project. The root cause of that was the design of the solution.

Appendix E: Review of IS Research Method

E.1 Overview

In a paper by Hilary J. Holz, Anne Applin, Bruria Haberman, Donald Joyce, Helen Purchase, Catherine Reed (2006), they argued that in computer science, research methods have historically been passed from advisor to student via apprenticeship. There is a lack of consensus on the nature of Computing Research Methods (CRM). They argued that the lack of formality of CRM has been a big issue for students and teachers as well. They proposed a CRM teaching theory separating the skills into two categories and addressing respectively. The two skill sets are core skills and specific skills. Core skills include organisational, expressive, cognitive and meta-cognitive skills, which are basic any research activity. Specific skills include those more specific to research project and the computing field, such as specific research method, collecting and analysing data and result presentation. However, the review has found few theories and categorizations on research method for computing and information system.

E.2 Classification

.E2.1 Epistemology Classification

According to a paper by Roelien Goede, Carina de Villiers (2003), information systems is an inter-disciplinary field of scholarly inquiry, where information, information systems and the integration with the organisation is studied in order to benefit the total system (technology, people, organisation, and society). Information Systems research can be classified as positivistic, interpretive or critical research.

Positivistic Research

A positivistic research is to gather evidences for formal propositions. It also includes quantifiable measures of variables, hypothesis testing. Finally, it should draw up conclusions about a phenomenon, which is from a representative sample to a stated population.

Interpretive Research

An interpretive research is to try to give explanations or to understand phenomena through accessing the meanings participants assign to them. In interpretive research, it is assumed that our knowledge of reality is gained only through social constructions

such as language, consciousness, shared meanings documents, tools, and other artefacts. Interpretive research tends to be subjective and focuses on the complexity of human sense.

Critical Research

Critical research is to aim for making changes and helping to eliminate the causes of unwarranted alienation. It presumes that everything and everyone has unfulfilled potentiality, by critically evaluating the reality, identifying restrictive conditions and recognizing new possibilities, it will enhance the opportunities for realizing human potential.

E.2.2 Interpretive Research

The Categories

In a paper by M. R. de Villiers (2005), which was focused on interpretive research, the author proposed three categories of interpretive research. They are development research, action research and grounded theory.

Development Research

Development research is also called research with a development goal. It aims to develop practical and innovative ways of solving real problems. It is also for proposing general design principles to inform future decisions.

Action Research

The action research approach emanates from the behavioural sciences and encompasses a variety of research and intervention methods. It was originated in action-based social psychology. It was argued by its founder Kurt Lewin of the University of Michigan that complex real social issues could not be investigated within laboratory. It was argued that action research is suitable for investigating the introduction of technologies into any organizations.

Grounded Theory

Grounded theory is an approach in which theory and models are generated inductively from the analysis of contextual data. Grounded theory (GT) involves the discovery of concepts and hypotheses as theory emerges from data. There is no testing or replication of a priori theory. Like action research, it has roots in social science, but specifically in sociology. GT should account for variation in domain behaviour by defining categories, properties and relationships. The data may be quantitative, qualitative or a combination, but in the case of qualitative, it is essential that it be systematically collected,

analysed and coded. Methods used in GT must be carefully selected or defined, so that a grounded theory emerges systematically and inductively through covariant ongoing collection and analysis. GT research can be adjusted, expanded, and refined during its process.

The Principles

In a paper by Klein and Myers (1999), the authors proposed a set of principles for conducting and evaluating interpretive field studies in information systems. These principles are summarized as followings:

- 1) The fundamental principle of the hermeneutic circle

This principle says that all human understanding is achieved by iteration between the interdependent meaning of parts and the whole they form. The iteration may consist of many circles. This principle is fundamental to all the other principles

- 2) The principle of contextualisation

This principle requires critical reflection on the background of the research setting, so the researcher could see how the current situation under investigation emerged.

- 3) The principle of interaction between the researchers and the subjects

This principle requires a critical reflection on how the research materials (or 'data') were constructed by the researchers or the participants. It recognizes that the participants could be just as like the researcher. This principle is lessened if the researcher does not interact with the participants but only deal with document or historical data.

- 4) The principle of abstraction and generalization

This principle requires relating the idiographic details revealed by the data interpretation to the theoretical, general concepts. In other words, the research should relate particulars described using the previous principles to very abstract categories; unique instances can be related to concepts that apply to multiple situations.

- 5) The principle of dialogical reasoning

This principle requires sensitivity to possible contradictions between the preconceptions and actual findings with subsequent cycles of revision. In other

words, the researcher must confront his or her preconceptions formed before the research with the data that emerge through the research process. The researcher should make the historical ideas as transparent as possible to the reader and himself or herself.

6) The principle of multiple interpretations

This principle requires the researcher must be aware of possible differences in interpretations among the participants. The researcher must examine the influences that the social context has upon the subjects and seek out multiple viewpoints along with the reasons for them. Moreover the researcher should confront the contradictions with each other, and revise his or her understanding accordingly.

7) The principle of suspicion.

This principle requires sensitivity to possible ‘biases’ and systematic ‘distortions’ from the participants. Even though previous principles would have helped with various forms of critical thinking, and a research would be more interested in finding out the truth than discovering “false preconceptions.”

E.2.3 Survey of IS Research Method

In a paper by Robert L. Glass, V. Ramesh, Iris Vessey (2004), they had done a survey and summarized current research methods in Computer Science (CS), Software Engineer (SE) and Information System (IS) in the following table:

Research Method		CS	SE	IS
AR	Action Research	-	0%	0.8%
CA	Conceptual Analysis	15.1%	43.5%	14.7%
CAM	Conceptual Analysis/Mathematical	73.4%	10.6%	12.1%
CI	Concept Implementation (Proof of Concept)	2.9%	17.1%	1.6%
CS	Case Study	0.2%	2.2%	12.5%
DA	Data Analysis	0.2%	2.2%	5.3%
ET	Ethnography	-	-	0.2%
FE	Field Experiment	-	<1%	1.6%
FS	Field Study	0.2%	<1%	24.5%
GT	Grounded Theory	-	<1%	0.2%
HE	Hermeneutics	-	<1%	-
ID	Instrument Development	-	-	3.5%
LH	Laboratory Experiment - Human Subjects	1.8%	3.0%	16.2%
LR	Literature Review/analysis	.3%	1.1%	0.8%
LS	Laboratory Experiment - Software	1.9%	<1%	0.6%
MP	Mathematical Proof	2.4%	<1%	0.2%
PA	Protocol Analysis	-	-	1.2%
SI	Simulation	1.8%	1.1%	1.4%
ES	Descriptive/Exploratory Survey	-	1.6%	2.7%

Table E. 1 Table by Robert L. Glass, V. Ramesh, Iris Vessey (2004)

Some of those most common methods listed in the above table are defined as following according to a paper by Hilary J. Holz, Anne Applin, Bruria Haberman, Donald Joyce, Helen Purchase, Catherine Reed (2006):

Action Research: *“a specific research approach in which the researcher generates new social knowledge about a social system, while at the same time attempting to change it.”*

Conceptual Analysis/Mathematical: *“to facilitate the classification of research that utilizes mathematical techniques.”*

Case Study:

- *“Single Case: examines a single organization, group, or system in detail; involves no variable manipulation, experimental design or controls; is exploratory in nature.*
- *Multiple Case Studies: as for single case studies, but carried out in a small number of organizations or context.”*

Field Study: *“no manipulation of independent variables, involves experimental design but no experimental controls, is carried out in the natural settings of the phenomenon of interest.”*

Grounded Theory: *“to develop a theory from data rather than gather data in order to test a theory or hypothesis. This means that qualitative methods are used to obtain data about a phenomenon and that a theory emerges from the data.”*

Laboratory Experiment: *“manipulates independent variable; controls for intervening variables; conducted in controlled settings.”*

Laboratory Studies: *“examination of computer-organization problems within a research goal setting of acquiring knowledge that is separate and distinct from the normal operational goals of the organization under study.”*

The authors argued some of those popular research methods in the Information System (IS) areas included Field Study, Laboratory Experiment, Conceptual Analysis, and Case Study.

E.3 Quantitative & Qualitative Research

E.3.1 Definition

Quantitative Research

According to a paper by Michael D. Myers (1997), it was originally developed in the natural sciences to study natural phenomena. It is now well accepted in the social sciences include survey methods, laboratory experiments, formal methods (e.g. econometrics) and numerical methods such as mathematical modelling. It is particular

popular in marketing research and social events such as general election. According to a paper by Hilary J. Holz, Anne Applin, Bruria Haberman, Donald Joyce, Helen Purchase, Catherine Reed (2006), in a simple term, quantitative research is a process of collecting of numerical data in order to explain, predict and/or control phenomena of interest.

Qualitative Research

According to a paper by Michael D. Myers (1997), it was developed in the social sciences to enable researchers to study social and cultural phenomena. Examples of qualitative methods are action research, case study research and ethnography. In qualitative research, research data is collected in observation and participant observation (fieldwork), interviews and questionnaires, documents and texts, and the researcher's impressions and reactions. By processing the data, researchers can then explain, predict and/or control phenomena of interest. Human factors play important parts in any qualitative research on social issues, including areas of information system.

E.3.2 Discussions

In a paper by Orit Hazzan, Yael Dubinsky, Larisa Eidelman, Victoria Sakhnini, Mariana Teif (2006), the author argued the qualitative research was usually used for the investigation of social phenomena, which involved people and different kinds of processes. Qualitative research was also usually conducted as case study. In case study, we would learn about environments, situations and processes, we might not be able to do so by quantitative data analysis. Quantitative data analysis could be useful in many situations and would help to argue for generalization, but it would not enable the researcher to explore all aspects of complex situations. On the other hand, qualitative research was often used to enable the researcher to understand and explore complex situations from many different angles. Qualitative and quantitative methods are not mutually exclusive. Many studies require assorted inquiry methods to cover the terrain and provide triangulation. Another role of qualitative research is as exploratory work, setting the foundation for quantitative research. Qualitative findings from basic research in new areas can be used to formulate hypotheses and questions for quantitative analysis, which then yields empirical results to test, verify and extend the qualitative hypotheses.

In a paper by M. R. de Villiers (2005), it was argued most research methods could be placed on a Positivist – Interpretive axis, tending from the quantitative to the qualitative, overlapping with each other. The following figure extracted from the paper summarizes the theory in a graphical way.

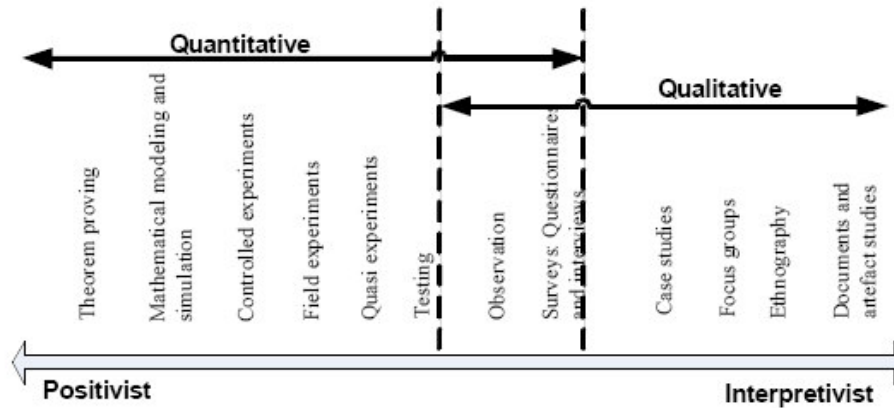


Figure E. 1 Figure by M. R. de Villiers (2005)

E.4 Case Study

E.4.1 Overview

The term "case study" can have multiple meanings and is used in far too many places than it should be for research purposes. In many cases, it is used to describe a unit of analysis (e.g. a case study of a particular organisation) for marketing purpose for new products. The discussion here concerns the use of the case study as a research method only. According to a paper by Michael D. Myers (1997), case study research is the most common qualitative method used in information systems. Although there are numerous definitions, Robert K. Yin (2002) defined the scope of a case study as follows:

“A case study is an empirical inquiry that: investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.”

As the above definition of case study implies, most case studies are of qualitative research trying to investigate and understand complex issues within their context. There are case studies using quantitative methods depending on their data collection methods. Case study can also be of positivist or critical research, depending on the underlying philosophy used by the researcher. Some scholars such as Yin (2002) and Benbasat et al. (1987) are advocates of positivist case study research, whereas others such as G. Walsham (1993) are advocate of interpretive in-depth case study research.

E.4.2 Case Study in Information System

According to a paper by Michael D. Myers (1997), clearly, the case study research method is particularly well-suited to IS research. He argued that the object of a case study for information system is within the organization; it is most likely to be organizational rather than technical issue. Most information system researches are studies of some technologies and their related procedures in the context of organizations. Case study method has clear advantages in such settings. First, case study provides multiple perspective and it is based on a specific context. Most information systems research can product multiple results and are bound to their context within the organization, without such context, any research would be hard to carry out and its result would be hard validate. Second, case study collects data using multiple methods, such as through observation, interview and document. For any information system search, it is important multiple sets of data are collected and analysed, as almost all information technologies and their systems have multiple aspects and impacts on the organizations, such as their people, their processes and the input/output results.

According to Dewayne E. Perry, Susan Elliott Sim, Steve Easterbrook (2006), they argued for case studies to be used for software engineers. They argued that case studies are a powerful and flexible empirical method, which is also applicable to research for information system. Most case studies are used for exploratory investigations, both prospectively and retrospectively. They are targeted for providing explanation or constructing a theory. They can also be used in the validation of research results. Because they are such flexible and adaptive research methods, they have become popular information system research. Case studies are often used in papers to understand, to explain or to demonstrate the capabilities of a new technique, method, tool, process, and technology.

Methods are heavily influenced by the aims of the research and the specific questions that need to be answered. Case study approach is appropriate because 'it is a research strategy, focusing on understanding the dynamics present within single settings' (Eisenhardt 1989). The research focuses on the contemporary phenomena, and in the context of IS research, the focus has been moved from technical to managerial and

organizational questions, and as a consequence, the research will investigate how context interacts IS applications. In order to understand the phenomenon, the research focuses on organizational structures, the business processing taking places etc (Benbasat et al., 1987).

Case study research is one of the most common approaches applied in IS field (Alavi et al., 1992), because it has multiple perspectives which are rooted in a specific context and provides multiple data collection methods (Richie et al.,2003). Therefore, case study research is not exclusively concerned with qualitative methods as all evidence will go to the data collection (Gillham, 2000). In addition, a case study approach is well-suited to IS study because the nature of the discipline is the study of IS as the technology per se in the context of organizations (Myers, 1997).

Many of the papers identified in the previous section used case study as method, especially on the topic of system integration and critical success factor.

E.4.3 Case Study Design

Case studies are an empirical method in their own right, with established design principles. To avoid pitfalls and achieve desirable results for case studies, there are some issues that must be addressed properly. First, a unit of analysis must be defined and selected, even though there may be no clear boundary sometimes. Second, the data collection must include those from all affected within the organization, especially when we have no control of the settings and variables. Last, the results of the research or any theory arising from it must be validated within the research. In summary, case studies must follow rigid rules and employ solid methods in order to produce useful results.

Elements:

Elements of a comprehensive case study should include the following as introduced by Yin (2002).

1) Study's Questions

The questions should start with 'who', 'what', 'where', 'how' and 'why'?

Those 'how' and 'why' questions would be the most appropriate for the case

study strategy, as they would lead to answers to allow the researcher to understand the complex case situations.

2) Propositions

There should one if there are possible outcomes of the research were expected by the researcher. However, a case study may not have a proposition if it is a purely interpretive research, as the objectives would be solely on understanding the case and discovery of new facts and relations.

3) Unit(s) of Analysis

The case study should clearly define the object/phenomenon under investigation.

4) Analysis Techniques

They are the logic links from the data to the propositions or relations for the case study. The techniques are to be employed to analyze evidence (collected data) and lead to the study's propositions.

5) Criteria for Interpreting the Findings.

They are measures for making sure the findings support the propositions. They are important for the quality of the case study and the results.

Quality Test:

The following is a table for quality test designed by Yin (2002)

Test	Case Study Tactic	Phase of Research
Construct Validity	Use multiple source of evidence	Data collection
	Establish chain of evidence	Data collection
	Have key informants review draft case study report	Composition
Internal Validity	Do pattern-matching	Data analysis
	Do explanation building	Data analysis
	Do time-series analysis	Data analysis

Appendix E: Review of IS Research Method

External Validity	Use replication logic in multiple-case studies	Research design
Reliability	Use case study protocol	Data collection
	Develop case study data base	Data collection

Table E. 2 Table by Yin (2002)

Validity

1) Internal Validity

This is related to issues that may affect the causal relationship between treatment and outcome. In other words, internal validity concerns whether the study has demonstrated a meaningful relationship between the variables under investigation. The threats would be from history, testing, maturation, and selection bias.

2) External Validity:

In social study, it concerns the population to which the results of a research study can be legitimately generalized. In other words, it is about generalization of study result from the samples included in the study using inductive logic. Many external validity involves inductive speculation that would go beyond the limits of the data and the research design. Threats would be from interaction of selection and treatment, and interaction of history and treatment.

3) Construct Validity

This concerns the logical confidence one can have in whether the variables in a study are valid measures of the corresponding constructs in the theory being tested.

4) Conclusion Validity:

This criterion concerns the validity of statistical conclusions drawn about the relationships among the variables investigated in the study. Threats to conclusion validity arise from a lack of statistical calculations or misuse of statistical assumptions that lead to incorrect conclusions by the researcher.

In summary, there are interrelationships between the four validity criteria.

- 1) Internal validity is the primary validity; all other validities would become meaningless without it.
- 2) A study would have external validity but weak construct validity.
- 3) A study would have construct validity but weak external validity.
- 4) Conclusion validity is a special case of internal validity.

Key Issues

In a paper by Graeme Shanks, Anne Parr (2003), a set of key issues for positivist case study research was identified and agreed by the authors. They are summarized as followings:

- 1) Concepts in the theory should be well defined
The author of the paper did not propose a “theory” but simply formulate some propositions. However, there is an implicit theory which would have exists with the propositions, which have been made clear before the case study.
- 2) Propositions should be converted to hypotheses before they are empirically tested
Propositions are predictions that may be deduced logically from theory. Values of concepts in propositions should be operationalised before testing can be conducted. Each of the values should be assigned an indicator. These indicators are then used in the proposition to form and test hypothesis. Once hypotheses have been produced, they may be used in the studies. Direct testing of propositions weakens the rigour of the study and produces inaccurate findings.
- 3) The boundary of the theory should be consistent with other referenced studies
The boundary of the case study need to be defined consistently with the scope of others referred in the study. When synthesizing propositions from the literature, the researcher should ensure that the scope and the boundary of the studies are the same or very similar.
- 4) A single case is not the same as a single experiment
The researcher would compare a single case studies to experiments, but must be aware that the output of such experiment might not be generalisable beyond the set of circumstances in the experiment. Additional experiments with different

circumstances would confirm the theory strengthen and extend the generalisability of the theory. However, carefully designed single case study would still be generalisable.

5) In a single case, strong hypotheses can be readily refuted

For hypotheses that involves causal interactions, they would be easily refuted by selecting specific case situations. However, if only case produces disconfirming evidence in multiple positivist case studies, the hypothesis need not be changed then.

6) Consider a move to post-positivism

Post-positivism takes into account our imperfect knowledge of reality and our inability to be objective. Propositions therefore should allow for “uncertainty” about them to a degree, particularly in the associations between concepts.

More Case Study Examples

The review had identified numerous papers using case study as method, apart from those which already been reviewed in the previous sections. Some of them are summarized in the followings:

In a paper by Bente Anda, Kai Hansen (2006), the author did a case study on the application of UML in legacy development. They argued case studies were practical and effective if we were to understand the actual practice of software development. Such understanding was an essential prerequisite to software engineering research. A major strength of the case study approach was that it would allow the study of a phenomenon within its real-life context. The author conducted a case study in a large project during a software process improvement initiative with focus on UML-based development in a large company. In the project, some of the development teams maintained existing components, while other teams designed and developed software from scratch. Through the case study, the authors argued that those who applied UML in modelling and maintaining legacy software experienced more challenges and fewer benefits using UML than those developers who used UML for developing new software. In the end, they argued there was a need for better support on applying UML in legacy development.

In a paper by Hee-Woong Kim, Shan L Pan (2006), they did the multi-cases study on process model of information systems implementation for customer relationship management systems. The authors argued because there was a lack of existing empirically based models on the process of IS implementation, they adopted a qualitative multiple case study approach. They had selected three cases to provide examples of polar types: one was successful implementation and two were failure. They selected the cases with other criteria including common characteristics of the cases, such as business context, scope of systems and life cycle type in the development process. The case study examined the process of IS implementation by explaining how factors influence each other and how those interactions produce results. Based on one successful case and two unsuccessful cases, they developed the process model of IS implementation. The process model explained how and why a set of factors influences each other and lead to certain outcomes in IS implementation. They argued the results complemented the findings of other variance-based research. Their case study of the proposed process model has revealed a predictive capability. In the cases, it was revealed one change in any factor in the process model would set off other changes through all other related factors. The cases also revealed that at any point in the course, alternative paths or corrective actions could be taken. The case study had facilitated an understanding of how to reverse repeating patterns of IS failure. It also would be used for new projects. They argued the process model, as a result from the case study, could be translated into development strategies and tactics.

E.5 Triangulation

E.5.1 Overview

In a paper by Michael Quinn Patton (1999), it was argued the term "triangulation" is taken from land surveying. With two landmarks you could take bearings in two directions and the intersection point could be exactly located. It was argued to be more accurate than using a single landmark to locate a point somewhere along a line in a certain direction from the landmark. It was also argued the notion of triangulating was adopted metaphorically from the world's strongest geometric shape the triangle.

The concept has been used in many research fields, including social research. The logic of triangulation is based on the principle that no single technique ever adequately solves the problem and provides full explanation. Because each technique would reveal different aspects of the research target, multiple methods of data collection would provide more data and multiple analyses would provide more accurate results. Researches using only one method are more vulnerable to errors linked to that particular method than those using multiple methods, as different types of data would provide cross-data validity checks.

Triangulation within a qualitative inquiry would be combination of different kinds of qualitative methods, mixing purposeful samples, and including multiple perspectives. It would also be achieved by combining qualitative and quantitative methods. From Denzin (1978) and Patton (1999), four types of triangulation were identified and explained below:

1) **Method triangulation**

It is about checking out the consistency of findings generated by different data collection methods. In many research, qualitative and quantitative data are both included. They would reveal complementary aspects of the same phenomenon. It is common for a qualitative researcher to discover the most insights in research with both qualitative and quantitative method.

2) **Triangulation of sources**

It is about examining the consistency of different data sources from within the same method. For example, the research compares opinion gathered from interviews with those discovered in document.

3) **Analyst Triangulation**

It is about using multiple analysts to review findings or using multiple observers and analysts. It would reduce possible perception or blind spots of an individual analyst.

4) **Theory/perspective triangulation**

It is about using multiple theoretical perspectives to examine and interpret the data

In summary, by combining multiple observers, theories, methods, and data sources, researchers can improve the quality and reliability of their study. However, triangulation would be very expensive. A researcher's ability and often limited budget

or short time frame could affect the amount of triangulation that is practical, especially combinations of interview, observation, and document analysis would only be achieved with a lot of fieldwork.

There are some common misunderstandings about triangulation. They were pointed out by Patton (1999). They are:

- 1) The goal for analyst triangulation is not to seek consensus, but to understand multiple ways of seeing the data.
- 2) The goal for data source triangulation is not to seeking essentially the same result, but to test for such consistency. An understanding of inconsistencies across different kinds of data can lead to deeper insight into the relationship between inquiry approach and the phenomenon under study.

Other theories for categorizing triangulation were also identified. For example, triangulation would be categorized into:

- 1) Methodological Triangulation.
 - a. One analyst using two or more research methods.
 - b. Two or more analysts using the same research methods.
 - c. Two or more analysts using two or more research methods.
- 2) Theoretical Triangulation. For example, a research could study the behaviour of a social group from both a Structuralist and an Interactionist theoretical perspective.

E.5.2 Triangulation in IS Research

For IS research, triangulation has been adopted mainly as using multiple data sources in an investigation to produce understanding. Some see triangulation as a method for corroborating findings and as a test for validity. As argue by Cohen D, Crabtree B. (2006), this would be controversial as it assumed that a weakness in one method would be compensated for by another method, and it would always possible to make sense between different accounts. It was suggested that rather than seeing triangulation as a method for validation or verification, qualitative researchers should use this technique to ensure that an account is rich, robust, comprehensive and well-developed.

In a paper by Lynne Johnstone (2007), the author argued for the following points:

Why is it used?

While qualitative research methods are employed, more and more researchers use triangulation in data analysis. The term was originally from navigation where it referred to the point of convergence of various navigation points. In the area of IS research, it means using a combination of methods in a study of a phenomenon. If multiple sources of evidence are used, the research would neutralise any bias in a particular data source, investigator and method. As result, researchers can have greater confidence in the conclusions.

How should it be used?

Triangulation was originally used for combining rationalistic and naturalistic paradigms, but it is now more used in naturalistic only studies. It is also argued that triangulation should only be used when data from one source is used to support data from another, and there is such need in the study. It may not be as simple as its definition, because all the evidence rarely, if ever, converges on 'the fact'. Non-convergent data would be the completeness or expansion of the 'fact'. There is really no clear guideline as how researchers might handle conflicting evidence in their research projects.

The author argued the logic of triangulation in research is essentially the same as that employed in legal proceedings, when a defendant is judged guilty or not guilty on the basis of evidence provided. In research, triangulation should be used to help us resolve the questions about contradictory evidence and data of multiple types and from multiple sources.

E.5.3 Triangulation in Case Study

It was described as in relation to case studies involving multiple sources and/or types of evidence converging on a 'fact' by Yin (2002). Triangulation is commonly referred to as making use of multiple source of evidence in a case study. The reasons argued by the author were:

- 1) It was one of the major strength of case study as case study generally could provide a lot more different types of evidence than other research method.
- 2) It would not only allow the researcher to address a boarder range of issues, but most importantly, to develop converging lines of inquiry.

- 3) It would be used to address construct validity, as a case study using multiple source of evidence would be rated more highly than those relying on a single source of information.

There are also issues regarding triangulation for case study, as argued by Yin (2002). They were:

- 1) The collection of multiple evidences would be more expensive than if data were collected in a single source.
- 2) The researcher needs to have the skill and technique of variety of data collection.

In summary, triangulation is one of the three principles for case study as argued by Yin (2002). However, it would also impose a great burden to the researcher.

E.6 Summary of IS Research Method

The review found a large number of papers on the topic of IS research methodology. There were a large number of IS research methods which would be categorized by different ways. The major categories are positivistic, interpretive and critical research. Research would also be quantitative or qualitative based. From a survey done in 2004 found in the review, there were certain methods used by most IS research, including field study, laboratory experiment, conceptual analysis, and case study. The same survey was also done on Computer Science and Software Engineering. There were different research patterns in different fields. The choice of research method would really depend on the research topic and target environment. The review had also summarized some basic elements and consideration that would affect the quality of a research. Since the review had found many papers using case study in the previous chapters, more attention was given to the case study method for finding out why it would be chosen and how it would be done. Many more papers using case study were identified and looked at. Theories of triangulation were also given some special attention, as they are hot topics in IS research and are important technique for improving reliability of a IS research.