An Investigation into the Technological and Organizational Factors that Contribute to the Successful Implementation of Global Distributed Infrastructures

By

Rajive B. Jain

A dissertation submitted to the Graduate Faculty in Business in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2004

UMI Number: 3115261

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

This manuscript has been read and accepted for the Graduate Faculty in Business in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

1.31.04
Date

Dorothy G. Dologite
Chair of Examining Committee

January 31/04
Date

Executive Officer

Dr. Stephen Gould

Dr. George Schneller

Supervisory Committee

The City University of New York

ABSTRACT

An Investigation into the Technological and Organizational Factors that Contribute to the Successful Implementation of Global Distributed Infrastructures

by

Rajive B. Jain

Adviser: Professor Dorothy G. Dologite

This dissertation presents the findings of a grounded theory-based approach to study the implementation of three generations of middleware technology standards in large global organizations. The findings are used to develop a theoretical framework that identifies the key organizational and technological factors that influence the successful implementation of these infrastructures.

The research reveals that lack of adoption of middleware technologies has more to do with broader industry-wide technological factors than those concerned with the internal organization of the firm. Specifically, a dominant design has not yet emerged in the middleware arena and that has led to frequent technology changes which have negatively impacted the implementation and use of these technologies.

This research extends the current models of technology adoption and implementation by adding another dimension, maturity of technology, which can have a significant impact on the success of a technology in a firm. This study also demonstrates that the findings in the technology innovation discipline can be applied to middleware

technology as well, and that middleware technology is a specific case of the generalized models developed in that area. It also enhances the findings of the technology innovation literature by identifying middleware-specific factors that impact the emergence of a dominant design.

This has important implications for researchers and practitioners. For researchers, it is recommended that they explore other evolving contemporary IS technologies to identify technology-specific forces that impact their success, develop strategic frameworks that can help to evaluate the implications of using such technology, and identify factors that influence vendor strategies. For technology mangers, it is recommended that they take into account the state of the technology to avoid the possibility of a lock-in into an unstable technology that may lead to expensive migrations, design solutions that are product independent, monitor their vendors by paying close attention to their strategies, and participate in the standardization process. For middleware vendors, it is recommended that they develop middleware technology with integration and interoperability in mind, because that is the primary feature that users value, and that, if they are engaged in the middleware technology standardization process, they encourage all major vendors participate.

ACKNOWLEDGEMENTS

My long journey from a small, sleepy industrial town in India to the fast moving corridors of Wall Street could not have been made without the contributions of many individuals. This dissertation is just one more milestone that these individuals have helped me achieve. The following are some of the key individuals whose contributions I wish to acknowledge.

Thanks to my father, late Mr. B .P .Jain, who gave me unconditional love and support throughout his life. He made me feel that everything was possible for me and that all I had to do was make up my mind to do it. His confidence in me still inspires me to dream and to make the big leaps. He showed to me that it is possible to succeed in this world by being a good human being. I bow my head in gratitude and admiration.

I especially want to thank my wife Priya, whose love, encouragement and support has made all my work possible. Her complete confidence in my ability to complete challenging assignments keeps me moving forward in life and was crucial in the completion of this dissertation. She has made good times more enjoyable and tough times easier to bear by maintaining perspective and balance in all aspects of our life. She spent countless hours taking care of the children, doing the household chores and providing us all with a comfortable and loving home while I toiled away trying to make my dreams a reality. She did all this while holding a full time job! Priya, I am blessed that I have you in my life.

Especial thanks are due to my two loving daughters, Shefali and Esha. Both of you were very understanding when I was not there because I was “working on the PhD”. Both of you are such a joy! I am blessed that I have you in my life.

Thanks to my mother, Mrs. Prem Lata Jain, who through her discipline and love, instilled in me a sense of responsibility and grounded my dreams in reality. She taught me that everything was possible for me, provided I worked for it. Her discipline guided my dreams and ensured that they are realized through intelligence and hard work. I am blessed that I have you in my life.

Thanks to my second set of parents, Mr. P. L. Kohli and Mrs. Saroj Kohli, who have, over the years, managed to keep alive my dream of completing my PhD. It was your encouragement and support that prevented me from "dropping out" at the weakest moments. You have helped us raise Shefali and Esha when I was not there for them. I feel lucky that I have you in my life.

Thanks to my sister, Sumedha, who taught me how to live life boldly and without fear. She taught me that in the face of a storm, it is best to stand up and walk boldly into it rather than hunker down and run for safety. She taught me that at times of grief, it is best to control your emotions and take care of everyone else. I am blessed that I have you in my life.

Poonam and Sandeep, thanks for your encouragement and support during the initial phases of this project. I remember I got so much done in those years because I was so relaxed with you folks around! I feel lucky that I have you all in my life.

Throughout my life I have been lucky to have mentors who served as role models and set me on a meaningful path. The following are some of my "friends, philosophers and guides" who have contributed extensively to my work.

Thanks to Mr. Ashok Sharma who spent countless hours with me during my childhood and college years helping me plod through math, engineering, and "growing up" problems all the while teaching me how to handle life. From him I learnt that confidence in nothing but the ability to think through issues and move forward in face of challenges. I am grateful for your love, affection, kindness and all that you have taught me.

Thanks to Mr. Naval Thakur who provided both emotional and financial support during my early years in the US. He taught me how to start over and over again despite failures. He is a unique individual who takes real joy in helping others achieve their goals. He is one individual I can count on for sound, practical and ethical advice when I really don't know what to do! I am grateful for your love, affection, kindness and all that you have taught me.

Especial thanks to Dr. Dorothy Dologite, my advisor and chair of the supervisory committee, for all her encouragement, advice and support. She is a gifted teacher and has the unique ability to create an environment of creative freedom, discipline and ease in the classroom. She spent many hours reviewing my work, providing feedback and encouragement. She inspired me to push myself to achieve higher levels of intellectual achievement. Thank you for being a wonderful teacher, an exemplary role model and a caring person. If I ever become a teacher, you will be the person I aspire to emulate.

Thanks are due to Dr. Joseph Weintrop and my committee members, Dr. Stephen Gould and Dr. George Schneller, for your advice, guidance and time.

Finally, I wish to acknowledge all of my friends who have listened to me complain, rejoice, laugh, cry, and ponder my way through these past years. Juliana & Stephan, Rashmi & Deepak, Richa & Sanjeet, Ruchi & Amit, Shikha & Hari - thanks for being there. Life is such a joy when you have friends!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The systems we provide to the firm must deal with continual changes to platforms, technologies, and architectures. New applications and platforms must interoperate with existing legacy systems. Virtual enterprises span multiple businesses. New implementation platforms, such as DCE, CORBA, J2EE, Web Services, .NET etc., are continually coming down the road, each claiming to be 'the next big thing.' Those who architect and build [computer] systems face many technology choices. How can we ensure that our systems are rooted in standards that will adapt over time to new hardware capabilities and software platforms, and that new and existing systems will continue to integrate and interoperate across changing platforms?

Managing Director of a global Fortune 500 firm, January, 2003

The use of Information Technology (IT) to gain competitive advantage, reduce operating costs and increase productivity is now ubiquitous in business. Over the past 15 years there has been a major shift in computing paradigms from centralized to distributed client-server computing. Driven by business needs, most organizations are building and deploying distributed applications and global distributed infrastructures in the hope of realizing greater strategic and operational gains.

However, these organizations are encountering a number of problematic technological and organizational issues as they move towards global distributed infrastructures (GDI), also known as middleware in the industry.

The chapter opening quote above quote aptly summarizes the daunting task technology managers in organizations face as they implement and use the middleware technology. These managers have to deal with frequent changes in middleware products and standards. They also have to integrate new middleware platforms with existing systems and evolve the legacy systems to take advantage of new functionality and features. At the same time, they must ensure getting a good return on their investment in technology.

Middleware technology has been evolving rapidly since its inception in the late 1980s. Figure 1 shows some of the key middleware standards and products that have been produced since the late 1980s. Appendix A explains all the middleware industry acronyms used in this figure and in the rest of this document. Some of these, like Microsoft's Component Object Model (COM) and Sun Microsystems' J2EE, are produced by individual vendors. Others, like Distributed Computing Environment (DCE), Common Object Request Broker Architecture (COBRA) and Web Services (WS), are produced by industry consortiums that try to promote them as industry-wide standards.

Since 1988, the middleware industry has experienced three generations of standards. The first two standards created by industry consortiums, DCE and CORBA, failed to gain a critical mass and were rejected by the user and vendor community. At present, the industry is in the process of creating the third standard, Web Services. In the meantime, several middleware products have appeared in the market. All these changes in technology have raised issues that concern organizations looking for middleware solutions. This research is an attempt to address some of the issues that plague the middleware technology users. This project investigates the technological and organizational factors that influence the success of middleware implementation in large

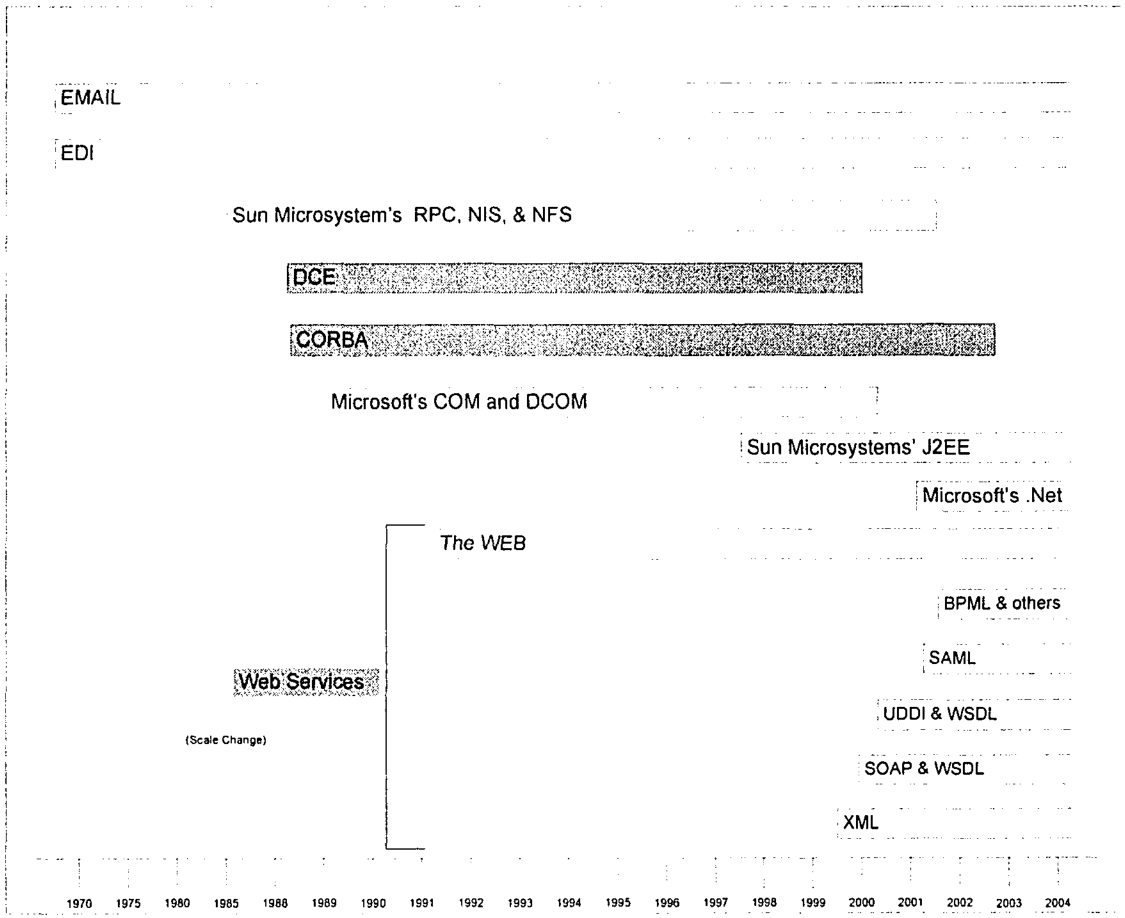

Figure 1: Middleware Standards/Products. Shaded areas show technology standards created by industry consortiums. These three standards are the main focus of this research. (Note: See Appendix A for a review of all middleware industry acronyms)

Source: Adapted from Ogbuji (2002).

global organizations. It also develops a theoretical framework that provides a deeper understanding of the issues to support both the user and research community. The following sections provide more details on this research project.

Throughout this document, the term middleware is used instead of global distributed infrastructures. Middleware is a term used heavily in the industry while distributed infrastructure is the term used in academic research in Computer Science and MIS. Since all the participants in this research were from the industry and used the term middleware, it is appropriate to substitute it for the more academic term.

The words "products" and "standards" are also used interchangeably in this document as the usage of these terms in the industry depends on the context. Microsoft's COM was referred to as a standard by Microsoft and it tried to promote it as a de facto industry standard because of its dominant position in the industry. Other vendors called it a product because it was not accepted as a standard by the rest of the industry. Similarly, Microsoft and Sun Microsystems referred to DCE as a product because it was not accepted by them as a standard.

NEED FOR THIS RESEARCH

The contemporary computer trade journals have discussed middleware issues extensively for over a decade - every year thousands of articles dealing with middleware are published. A survey, conducted as part of this project, of three leading industry analyst firms, Gartner, Giga, and Forrester, shows that they have published over 10,000 research bulletins on middleware since 1990.

The IS literature, on the other hand, has been surprisingly silent on middleware issues. A detailed review of literature, documented in Chapter 2, reveals a paucity of research on this topic.

Given the huge industry budget allocated for middleware technology purchases, $5.984 billion on Application Integration and Middleware (Correia et al., 2003) in 2002, and the high cost of implementation and use, it is important to understand why and how these technologies are used and what contributes to their success in organizations.

Because of the paucity of critical research on the subject of middleware, researchers have no conceptual tools or frameworks to guide a research effort. Practitioners, likewise, have no theoretical guidelines to support an understanding of the factors that impact the success of their middleware projects.

Research is required to fill this gap.

DESCRIPTION OF THE RESEARCH

This research addresses the following four open issues:

- the conspicuous absence of systematic research addressing the organizational and technological factors related to the implementation of middleware technologies in large global organizations;
- the resulting incomplete understanding of why the previous two generations of middleware technologies, DCE and CORBA, failed in the industry and what can be learned about the factors that will impact the success of the emerging new Web Services standard;

- the absence of a research foundation, specific to middleware technologies, that can assist researchers interested in investigating middleware and similar technologies; and
- the absence of conceptual frameworks, specific to middleware technologies, that can assist researchers and practitioners to better understand the factors that impact the success of their middleware projects.

This research project addresses the above issues by developing an explanatory and descriptive theory of middleware implementation in large global organizations. It develops an integrated framework that takes into consideration the organizational and technological context that shapes the strategies and actions of technology managers who are concerned about the success of middleware in their firms.

As the focus of this research is to understand the key factors that affect the implementation of middleware in large global organizations, the core research question guiding this study was "What are the critical technological and organizational elements that affect the implementation of middleware in large global organizations?"

To answer this question, the author conducted a grounded theory study of the experience of managers who have been involved with the implementation and use of the three generations of middleware technology in large global organizations – DCE, CORBA and Web Services. These technologies were selected for the following three reasons:

- Both DCE and CORBA were supported by a significant number of vendors in the information technology industry. A number of large global corporations, including the ones in this study, attempted to implement these products and failed. Failures provide signification insight into what went wrong.

- Web Services is in the process of being developed as a standard. Several components of the Web Services specification have been accepted by all the significant vendors in the middleware industry as a standard. The standardization process is in progress at this time and this provides a rare window of opportunity to study the evolution of a technology as it moves toward maturity.
- Since one of the primary uses of middleware is to integrate diverse technology platforms, it is important that the products are supported by the vendors who have a significant market share. In the middleware industry, only these three products produced by separate industry consortiums have had significant industry-wide acceptance.

The grounded theory methodology was used for this project because it is especially useful for generating concepts and frameworks in areas like middleware, where little or no research exists. Also, grounded theory methodology takes into consideration the impact of context, processes, conditions, and actions and interactions of key stakeholders – all the elements that are needed to study a complex technology in large global organizations.

In this research, an a priori theoretical perspective was not specified because in the grounded theory methodology, the substantive theory is "derived from the data, [that has been] systematically gathered and analyzed through the research process" (Strauss & Corbin, 1998, p. 12).

At the beginning of the project, an extensive review of technology adoption and implementation literature was conducted to gain perspective and to formulate initial questions that served as a "stepping off point during initial observations and interviews" (Strauss and Corbin, 1998, p.51).

As the data analysis progressed, it became apparent that the technological changes at the industry level, fuelled by forces of innovation, had more impact on the implementation of middleware in organizations than the internal technological and organizational factors.

At this stage, a comprehensive review of the technology innovation literature was conducted to integrate the empirical findings of this research with a related existing body of knowledge that has a pattern similar to what the author was finding in the collected data. This is a commonly accepted practice when conducting research without an a priori theoretical perspective, as is often the case with grounded theory methodology. See Robey et al. (2002) for an excellent example of research that follows a similar pattern.

Within the technology innovation literature, the concept of a dominant design drove the theoretical discussion of the findings from this research that are the subject of Chapter 5.

The empirical findings were used to develop a theoretical framework for conceptualizing the factors, both organizational and technological, that impact the implementation and use of middleware technologies. These findings are then integrated into the more generalized theoretical framework using the concept of "dominant design" from the technology innovation literature. The result is an empirically valid theoretical framework that can be used to understand the phenomenon of middleware implementation in large global organizations and to contribute to IS research as well as to IS practice.

CONTRIBUTIONS OF THIS RESEARCH

This research makes the following three contributions.

First, it generates an empirically valid model of the implementation of middleware technology in large global organizations and fills an important void in existing body of technology adoption and implementation research.

Second, this research extends the current models of technology adoption and implementation by adding another dimension, maturity of technology, which can have a significant impact on the success of the technology in organizations. The current technology adoption and implementation models are valid for stable technologies but fail to explain the implementation of technologies that are still evolving. This finding is an important contribution which can be applied to the implementation of other new emerging technologies in organizations like Wireless Global Wide Area Networks, Public Key infrastructures and Smart Cards.

Third, this research integrates and interprets the empirically generated model with an established framework from the technology innovation research stream. This enhances the generalizability of the model developed in this research and better enables researchers and practitioners to explain, anticipate, and evaluate the outcomes associated with the use of evolving technologies in large global organizations.

This thesis is structured as follows. The next section provides a brief overview of the three middleware technologies whose implementation was investigated in this research project. Chapter 2 provides an extensive review of the existing technology adoption and implementation literature. The purpose of this review is to identify the relevant concepts form past research and to formulate the questions that were used in the initial phases of this project. This also shows the gaps in the existing knowledge and provides a rationale for this grounded theory study. This is consistent with the guidelines recommended by May (May, 1986, p. 149). Chapter 3 describes the research methodology including selection of participants, data collection and analysis, and validation methods. Chapter 4 presents the findings including the model developed using the grounded theory approach. Chapter 5 is the discussion of the findings, and their implications for researchers and practitioners. This chapter focuses on the model developed in the previous chapter and further integrates it with the technology innovation literature to enhance its theoretical generalizability.

MIDDLEWARE TECHNOLOGIES OVERVIEW

This section provides a brief overview of DCE, CORBA and Web Services middleware technologies. Since industry consortiums have played a significant role in the development and success of these middleware standards, it is important to understand the role of these industry consortiums, their motivations for creating the standards, and the business and technical environment into which these middleware standards were conceived.

Since there is no organization or industry body that maintains a history of these technologies, the details in this section were obtained from a detailed review of industry analyst reports published since 1993 and from the web sites of the consortiums that created these standards.

DCE

In 1987-88, the Open Software Foundation (OSF) was formed by a group of technology vendors including Hewlett Packard (HP), International Business Machines (IBM), Bull, Digital Equipment Corporation (DEC), and others, to define a standard distributed environment for enterprise computing. Sun Microsystems and Microsoft were two prominent vendors who refused to join the OSF at that time.

The OSF model for technology standardization was to choose the "best of breed" technologies available in the market, integrate them into a common standard solution, and create a paper standard that would define the industry-wide standard distributed computing model.

OSF would also provide source code that would implement the standard. The member vendors would then port the code to their specific platforms and this would guarantee interoperability between the various platforms. The intent was to ensure that OSF did not repeat the mistakes of Open Systems Interconnect (OSI). OSI was an earlier industry consortium that developed standards which were widely accepted. Vendor offerings, however, were so different that products failed to interoperate.

OSF identified the following key areas of distributed computing as essential components of a standard distributed computing environment: communications protocol, directory services, security, file system, system threads, and time services. In 1988, OSF issued a "Request for Technology" to the computing community including technology vendors, educational and research institutions.

Bids were received from several sources including Apollo, HP, Sun Microsystems, DEC, Transarc, Massachusetts Institute of Technology (MIT), Carnegie

Mellon, and others. OSF evaluated the technology submissions and selected several solutions from various vendors in exchange for royalties. The selection process from the many industry bids received took more than a year. In 1989, OSF announced the winners. It did not select Sun Microsystems' technology for any of the DCE components. In retaliation, Sun Microsystems refused to join OSF.

IBM took on the task of taking the components from different vendors and weaving them into a homogenized solution that was christened Distributed Computing Environment or DCE. The first version of the code was released to the member vendors of OSF in 1991.

It took OSF another three years to produce an industrial strength version of DCE. In 1994 OSF released DCE 1.1 and this was considered to be the first usable version of DCE. This was the product that most end user organizations deployed. Table 1 summarizes the key events in the history of DCE.

Over the next five years, OSF developed enhanced versions of DCE but the technology failed to get widespread adoption in the vendor and end-user communities. In 1999, OSF, now called The Open Group, officially stopped the development of DCE standards.

CORBA

CORBA was created by the Object Management Group (OMG), an industry consortium founded in April 1989, with the aim of creating standardized object technology-based software. The eleven initial members included both technology vendors and end-user organization including 3Com Corporation, American Airlines, Canon, Inc.,

Table 1: A Brief History of DCE

Year	Event
1987-1988	OSF is formed
1988	OSF issues a "Request for Technology" (RFT) to the computing industry inviting bids for technology components that could be used in the standardized DCE middleware infrastructure.
1989	Key DCE component technologies are selected.
1990	The selected technologies are integrated by IBM into an integrated standard called DCE.
1991	OSF releases DCE v1.0 – first versions of DCE for the market
1993	Sun joins OSF but continues to promote its own distributed computing solution.
1994	OSF releases DCE v1.1. This version makes important improvements to security, administration and performance. This is considered to be the first industrial strength release of the product.
1996	OSF merges with X/Open, another industry consortium, to form The Open Group.
1997	With the popularity of DCE on the decline, the Open Group makes DCE source code freely available to the technology community.
1999	The development of DCE is officially stopped.

Source: OSF (2003)

Data General, Hewlett-Packard, Philips Telecommunications, Sun Microsystems and Unisys Corporation. Today, the consortium has hundreds of members.

OMG's model for technology standardization was to develop specifications that were then developed into products by vendors. These specifications were developed based on technology proposals from member vendors which were then analyzed, debated and modified in the OMG technical committees that were comprised of both vendors and end-user organizations. Once consensus was reached on specifications, they become adopted standards. These were then developed into products by vendors.

OMG released the first version of the CORBA specifications in 1991. The products based on these specifications were not commercially viable. OMG continued to develop the specifications further based on the feedback from the user community and CORBA 2.0 specifications were released in 1996. The products based on these are believed to be the first commercially viable version of CORBA. Table 2 describes the key events in the history of CORBA standards.

By this time it was evident that CORBA was not being adopted on a large scale by the end-user community. OMG started developing methodologies to support the design and development of object-oriented software with the hope that this would ease the task of migration to CORBA. In 1997, it formally adopted Unified Modeling Language (UML) as the standard for developing object-oriented software.

By the year 2000, the key vendor members of OMG had shifted focus to the development of Web Services. Though OMG still promotes CORBA, all its key members are actively involved with the development of Web Services standards.

Table 2: A Brief History of CORBA

Year	Event
1989	OMG is formed as a non profit organization to create vendor independent object technology components.
1991	CORBA 1.0 specifications are released with limited components
1992	CORBA 1.1 specifications are released. This is the first widely published version of CORBA. There are no commercially viable products yet.
1993	CORBA 1.2 specifications are released. This supports only the C programming language. This is still not a commercially viable product.
1996	CORBA 2.0 released – believed to be the first commercially viable specifications. Provides support for C++ and Small Talk.
1997	OMG adopts Unified Modeling Language (UML) as the standard for modeling systems.
1999	CORBA 3.0 specifications are released. OMG's focus on CORBA begins to fade as its key members begin to develop Web Services standards.
2000-2003	Though OMG continues to promote CORBA, most of the OMG members, particularly the influential vendors, are actively involved in the development of Web Services specifications. OMG is now focused on developing UML as a methodology for systems development and integration to address middleware technology issues.

Source: (OMG, 2003)

Web Services

Unlike DCE and CORBA that were conceived first as a concept by industry consortiums and then developed into standards and products, the idea of Web Services emerged gradually during the late 1990s as a result of several different efforts by various individuals, vendors and industry consortiums. Ogbuji (2002) provides an excellent history of the evolution of Web Services.

At present, the creation of Web Services standards is being driven by the following three key industry consortiums:

- World Wide Web Consortium (W3C)
- Organization for the Advancement of Structured Information Standards (OASIS)
- Web Services Interoperability Organization (WS-I).

Table 3 describes the key events in the history of Web Services development. Appendix A provides details on the history of these three industry consortiums.

Before 2001, various vendors and OASIS were developing specifications for different components of Web Services. In April 2001, W3C held a workshop to plan the future of Web Services, to outline the plans for the development of Web Services and to identify areas for further development. This event was described by industry analysts as the official beginning of the Web Services revolution (Ogbuji, 2002). At present, all the three consortiums are involved in the development of the Web Services standards.

Web Services architecture has been described as a "stack of services." Figure 2 shows the current state of the various services that comprise the standards. The services in the lower layers of the stack are now well defined and have been accepted as standard

Table 3: A Brief History of Web Services

Year	Event
February, 1998	The World Wide Web Consortium Issues XML 1.0 as a W3C Recommendation.
May, 1998	Dave Winer, working for Userland Software, introduces the idea of using a communication protocol based on Extensible Transfer Language (XML). This later develops into the communication protocol called Simple Object Access Protocol (SOAP).
1999	Sun Microsystems, working within the industry consortium Electronic Data Interchange (EDI) and OASIS, releases the specification for Electronic Business using Extensible Markup Language (ebXML). SOAP version 0.9 is released to the technology community for evaluation.
May 2000	IBM joins the SOAP bandwagon. It coauthors the 1.1 version of SOAP and submits it to W3C for consideration as a standard. IBM produces a Java language-based SOAP toolkit that is made freely available to the technology community. This launches the development of SOAP at the "grass roots level of the technology community" (Ogbugi, 2002).
March 2001	IBM, Ariba Inc., and Microsoft develop Web Services Description Language (WSDL) and submit it to W3C for consideration as a standard.
April, 2001	W3C holds a workshop to develop plans for the future of Web Services. This event is described by analysts as the official beginning of the Web Services revolution.
May, 2001	World Wide Web Consortium Issues XML Schema as a W3C Recommendation.
June 2001	World Wide Web Consortium Issues XML Base and XML Linking Language as W3C Recommendations.
October, 2001	World Wide Web Consortium Issues XML Information Set as a W3C Recommendation.
July 2002	W3C publishes first working draft of WSDL version 1.2.
June 2003	W3C issues SOAP version 1.2 as a recommended standard.

Source: Ogbuji (2002), W3C (2003), OASIS (2003) and WS-I (2003)

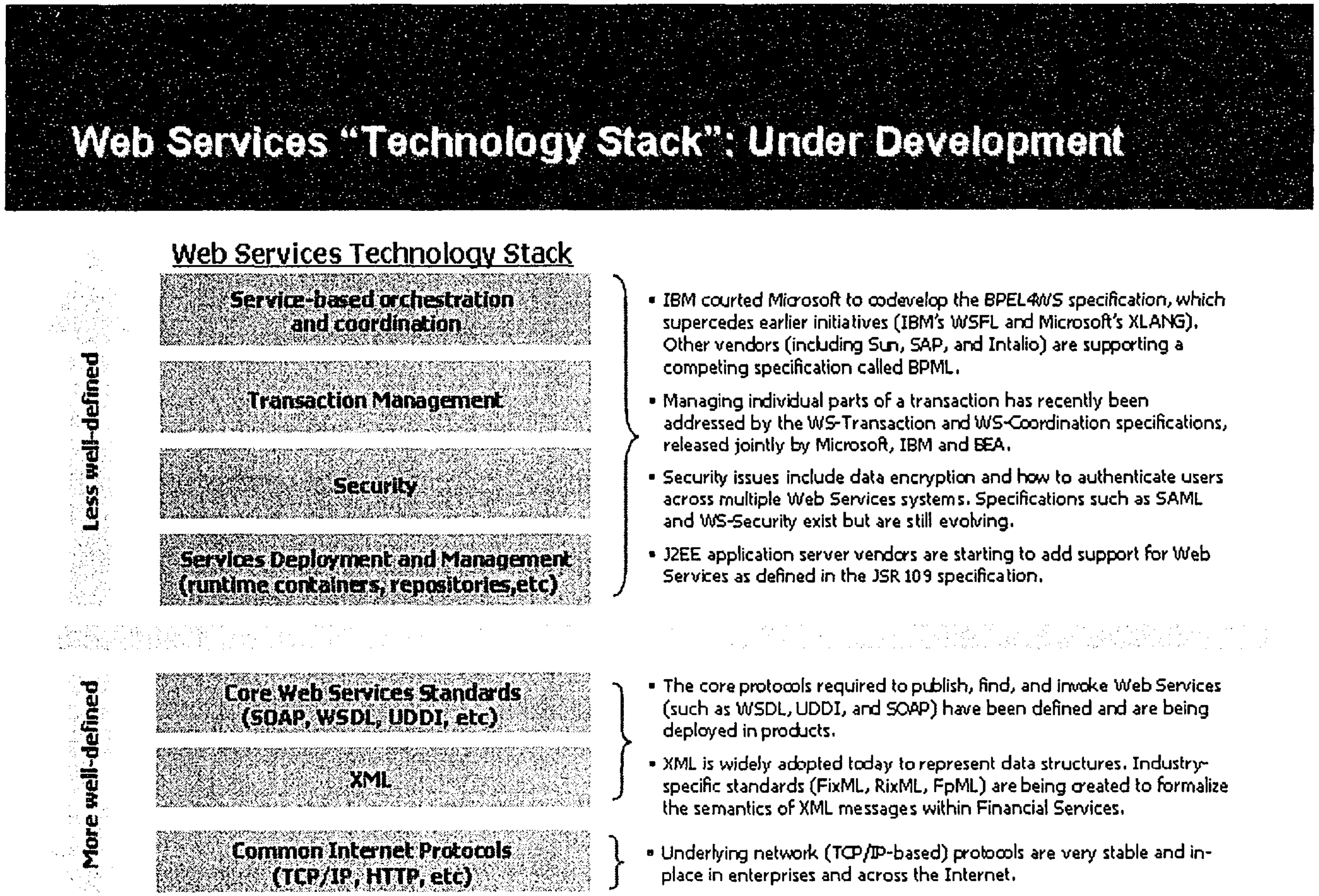

Figure 2: State of Web Services Standards (Note: See Appendix A for a review of all middleware industry acronyms)

Source: Internal strategy document made available by a research participant. Dated March, 2003 by W3C and other industry consortiums. There are products available in the market that meets these standards and are being used in the industry.

The battle for influence has now moved to the upper layers of the stack and the various vendors are producing competing standards that are beneficial to them and may give them a competitive advantage. Some of these standards are mutually exclusive andthe success of Web Services will depend upon the ability of the various consortiums to resolve these conflicts and produce a standard that will meet the needs of end-users. Figure 3 shows the competing standards.

Though the three consortiums work together and all the key vendors (HP, IBM, Sun Microsystems, and Microsoft) are members of all the consortiums, the exact working relationship between them is not clear. OASIS and WS-I categorically state that they are not standard setting organizations but promote interoperability and use of standards. W3C has a well defined process for standards setting.

Moreover, these consortiums frequently change direction and agendas depending on the power and status of the vendors who are on the board of directors. Appendix B provides detailed background on the three consortiums that are involved in the development of Web Services Standards.

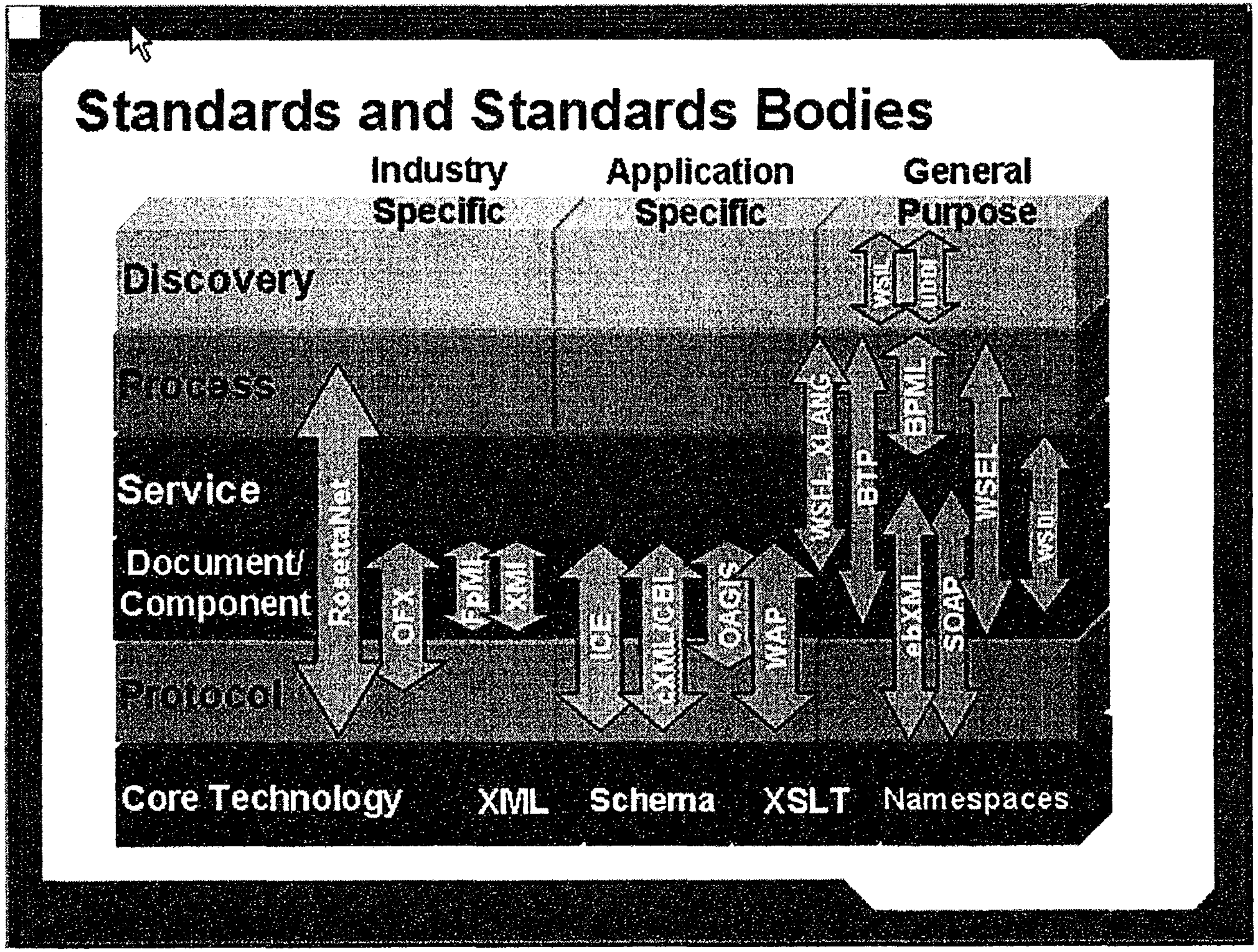

Figure 3: Competing Standard in Web Services

(Note: See Appendix A for a review of all middleware industry acronyms)
Source: Narsu (2002)

CHAPTER 2

LITERAUTRE REVIEW

During the last three decades, the complex issue of IT implementation in organizations has received considerable attention among researchers. These studies differ in their theoretical framework, the phases of the technology adoption and implementation, and in their treatment of implementation process, contextual factors and the interaction of the two. This section will focus on identifying major prior contributions in these areas. In addition, weaknesses in previous research will be identified. Finally, this analysis will be used to identify the gaps in the research and will be linked to the objectives of this study.

OBJECTIVES OF THE LITERATURE REVIEW

The primary objective of this study is to develop a grounded theory of the technological and organizational factors that impact the successful implementation of global distributed infrastructures in organizations. In the grounded theory methodology, literature review has a very specific intent and purpose that is different from that in traditional research projects. Technical literature was used for the following purposes in this research project:

1. It was used as a source of relevant concepts related to the successful implementation of information technologies. Strauss and Corbin (1998, p. 49) state that concepts from the literature can be used to make comparisons with the field data at the dimensional level thus enabling differentiation and more precise

specification of the emerging concepts. For example, in IT literature, terms such as "organization's understanding of technology", and "organizations commitment to technology" are strong concepts. Unfortunately, some of these concepts are not well defined and researchers use one or more specific properties of these in their empirical studies. In this review, such prevalent concepts are identified and their properties explored as part of the analysis.

2. It was used to identify relationships between concepts and to compare, refine, validate, explain, and differentiate the emerging relationships in the current research.
3. It was used to provide the background to formulate questions that were used in the initial phase of this project.
4. Once the data collection and analysis was completed, the findings of this review were "used to confirm findings and, just the reverse, findings...[were] used to illustrate where literature is incorrect, is overly simplistic, or only partially explains the phenomena. Bringing literature allows for extending, validating, and refining knowledge in the field" (Strauss & Corbin, 1998, p. 49).

Consistent with the grounded theory methodology, literature was reviewed throughout the data collection, analysis and theory development phases of this project.

LITERATURE REVIEW APPROACH AND SCOPE

After an extensive review of the literature it became apparent to the researcher that no articles related to middleware have been published in IS journals. On the other hand, thousands of articles on this subject have appeared in trade journals and magazines in the last decade. This indicates a significant gap between IS research and the changes in fundamentals of corporate computing brought about by the introduction of GDI. In addition, none of the research articles that the researcher reviewed addressed the global issues in technology implementation and deployment.

The past research in technology implementation is either application specific (e.g., MRP, EDI, ISDN) or related to end-user computing (e.g., ES, CASE, ESS, GDSS). This literature review focuses on a detailed analysis of the implementations of the key technologies with the aim of identifying the success factors related to these diverse system types that may also be applicable to GDI.

The domain of this literature review was derived from key IT implementation articles published in the following journals:

- Communications of the ACM
- Decision Sciences
- IEEE Transactions on Engineering Management
- Information and Management
- Information Systems Research
- Journal of Management Information Systems
- Management Science
- MIS Quarterly.

Technology implementation studies for the following key technologies were reviewed for this research:

- Computer Aided Software Engineering (CASE)
- Client Server technologies
- Database Management Systems (DBMS)

- Decision Support Systems (DSS)
- Electronic Data Interchange (EDI)
- Executive Support Systems (ESS)
- Expert Systems (ES)
- Group Decision Support Systems (GDSS)
- Information Centers (IC)
- Integrated Services Digital Networks (ISDN)
- Strategic Information Systems for Planning (SISP)
- Material Resource Planning (MRP)
- Open Systems.

The following sections identify the major contributions in the following broad areas in information systems research:

- theoretical models in technology implementation
- empirical research in technology implementation
- contextual factors relevant to technology implementation.

TECHNOLOGY IMPLEMENTATION THEORETICAL MODELS

The IS literature has reported several theoretical models that have identified the various dimensions on the phenomenon of successful implementation of technology. These include identification of the stages of the entire process, contextual factors that impact the various stages of the process, characteristics of the technology, users, and the fit between the task and the technology used to perform the task.

Researchers have recognized that technology adoption and implementation is a process that has some well-defined phases with distinct characteristics. Cooper and Zmud

(1990) outline a framework based on six implementation phases: initiation, adoption, adaptation, acceptance, routinization, and infusion. Initiation is the stage when IT solutions initiated in response to problems or opportunities. It ends when an IT solution is identified. Adoption is the stage of rationalization and political negotiations that ensures that all the stakeholders agree to back the chosen solution. It ends with the decision to invest the necessary resources to implement the solution. Adaptation is the stage during which the solution is developed, deployed and maintained. It ends when the solution is available to the end user. Acceptance is the stage when the organization members are encouraged to use the solution and it ends when the solution/application is used for day to day working of the company. At this stage, the solution is still an innovation and its use is not yet universal. Routinization is the stage during which the solution becomes part of the day-to-day activities of the organization. Infusion is the stage when the solution is used to its full potential.

Preece (1991) discusses another framework that consists of seven stages: initiation, progression, investment decision, planning and systems design, installation, operationalization and evaluation of the new technology.

Chengalur-Smith & Duchessi (1999) combine the two models and present a unified model in which the first two stages of Cooper and Zmud (1990) model are combined with the first three stages of Preece's model and tackled as "Initiation and Adoption" stage. This is defined as "the pre-implementation process of identifying and responding to a company's problems and/or opportunities, searching for appropriate IT solutions, and assessing the technology's benefits for management's approval." The other stages are classified as Implementation and Post-implementation stages.

This research will use the categorization developed by Chengalur-Smith & Duchessi and focus on success factors affecting the implementation phase of technology only. This means that the company has already made the decision to invest in a particular technology and the research studies the organizational and technological factors related to implementation. Implementation will cover "design, development, deployment, and availability to users and is being used in the day-to-day working of the company" (Cooper & Zmud, 1990).

Rogers' (Rogers, 1983; Rogers and Shoemaker, 1971) Innovation Diffusion Model (IDM) has been extensively adapted to the study of IT implementation. This model This theory posits that the success of innovation depends on the following five characteristics:

1. Relative advantage is the degree to which the innovation is perceived to be superior to the current practice in use.
2. Compatibility is a measure of the degree of fit of the new innovation with the existing environment.
3. Complexity is the degree to which the innovation is difficult to understand.
4. Trialability is the degree to which a potential adopter is allowed to experiment with the innovation prior to commitment.
5. Visibility is the degree to which the benefits of the innovation are visible to the stakeholders.

This model posits that higher relative advantage, compatibility, trialability, and visibility positively impact the adoption of innovation. Higher complexity negatively impacts the adoption of innovation. Tornatzky and Klein (1982) performed a meta-analysis of twenty-seven studies on innovation adoption and implementation and concluded that compatibility, complexity and relative advantage are consistently important. Rogers' innovation diffusion theory has been extensively used in various empirical studies applied to diverse information technologies including Integrated Services Digital Networks (ISDN) (Lai 1997), databases (Hoffer and Alexander, 1992), and Electronic Data Interchange (EDI) (Ramamurthy and Premkumar, 1995). Most of these studies have ignored the trialability and visibility factors of the theory. Various researchers have pointed out the deficiencies of Rogers' innovation diffusion theory when it is applied to organizational innovations because this theory was developed to predict the diffusion of mass-produced items. This theory does not take into account the environmental context, the organizational context and the characteristics of the tasks that the innovation impacts (Brancheau and Wetherbe, 1990; Cooper and Zmud, 1990; Zmud, 1982).

Kwon and Zmud (1987) identified five major classes of contextual factors that impact the processes and products associated with the adoption and implementation of technology: user, organization, technology, task to which the technology is being applied, and organization environment. User refers to the characteristics of the user community such as attitude to change, education level, job security etc. Organization refers to the characteristics of the organization in which the technology is being deployed and used e.g. centralized, formalized, specialized etc. Technology factors are the specific unique characteristics of the technology product being used or deployed. Task factors are the unique characteristics of the task to which the technology is being applied. For example, if the task is invoice processing, then this task may be characterized as "well defined" with no "task variety." Finally, the organization environment factors are those factors that are part of the organizations' operating environment such as uncertainty and inter-organizational dependence. Cooper and Zmud (1990) also identified the degree of fit between task and technology as a key factor that impacts successful implementation.

Davis' (1989) Technology Acceptance Model (TAM), based on theory of reasoned action, models the impact of user perceptions about the technology on their acceptance of it. The model posits that users' beliefs about usefulness and ease of use are of primary relevance for system acceptance behavior. Perceived ease of use also has a direct, positive effect on perceived usefulness. Both constructs have a direct effect on the acceptance of the technology. The two constructs can be affected by various external variables such as training, documentation and user support. Davis et al. (1989) used this model to explain the user's behavior across a wide range of technologies. Chau (1996) further classified usefulness as near-term and long-term and identified "implementation gap" and "transitional support" as two other external variables that can impact ease of use and usefulness.

Venkatesh and Davis (2000) extended the TAM, called TAM2, by adding theoretical constructs spanning social influence processes and cognitive instrumental processes. Subjective norm, voluntariness and image are the three key interrelated social forces impinging on an individual facing the opportunity to adopt or reject a new system. Subjective norm is defined as a "person's perception that most people who are important to him think that he should or should not perform the behavior in question." Job relevance, output quality, result demonstrability, and perceived ease of use are identified as the four cognitive instrumental determinants of perceived usefulness. This model postulates that both social influence and cognitive instrumental processes have a direct effect on user acceptance.

Chau and Tam (2000) developed a model based on the work of Zmud (1984) using the concepts of 'Technology-Push' (TP) and 'Need-Pull' to explain the adoption of open systems in organizations. Open systems is defined as "a comprehensive and consistent set of international standards and functional standards profiles that specify interfaces, services and supporting formats to accomplish interoperability or portability of applications, data and people" (Chau and Tam, 2000, p. 231). The Technology-Push construct proposes that technological advances trigger the sequence of events that end in diffusion of technology and its application. Need-Pull states that user needs drive the adoption of technology and innovations. This model posits that both TP and NP factors impact the adoption of technology. They identify the technology-push (lower costs and benefits) and need-pull (satisfaction level with current systems, market uncertainty, human resources and degree of formalization) factors that impact the adoption of Open Systems in organizations.

In summary, the Innovation Diffusion model identifies the characteristics of the technology that impact its adoption and implementation. Though this model is useful for products that are meant for "mass consumption", it is relevant for infrastructure technologies that are utilized across the entire firm and in that sense have a wide user base. The contextual factors identified by Kwon and Zmud (1987), and their influence on the different stages of the IT implementation process add another dimension to the complexity of the phenomenon. The TAM model focuses on the users' perceptions. These different models give different views of the phenomenon and researches have used a mix of factors from the various models to investigate their relationship to the success of the implementation. The next section provides details of the empirical research in this area.

EMPIRICAL RESEARCH ON TECHNOLOGY IMPLEMENTATION

Considerable empirical research across various technologies has demonstrated the validity of the theoretical models of technology implementation discussed in the previous section. This section describes some of the key empirical studies that demonstrate the validity and limitations of these theoretical models.

Lai (1997) and Lai and Guynes (1994) focused on ISDN technology and the organization characteristics to determine the critical success factors of ISDN implementation. Lai (1997) conducted a survey of ISDN implementation in two hundred and forty-seven companies in the US and demonstrated that greater compatibility, relative advantage and lower complexity increase the adoption of ISDN technology. The same study determined that organization characteristics like openness (measured as average number of external contacts), centralization (extent of participation in decision making) and formalization (extent of rule observance and job codification) positively impact the success of ISDN adoption. This is a direct validation of Rogers Innovation Diffusion Model for a product that is meant for "mass consumption" in an organization. Another validation of the Innovation Diffusion Model was provided by a longitudinal study of implementation of packaged software on mainframe by Cale and Eriksen (1994). They determined that over the short term, as viewed from the perspective of the end user, complexity, compatibility, and relative advantage impacted the implementation as predicted by the model. Over the long term, while these factors remained significant for continued success, it was important that the application also not be overly complex from the perspective of the support staff.

TAM has been validated in studies involving technologies that directly impact the end users. Numerous studies involving Expert Systems (ES), Decision Support Systems (DSS), and Computer Aided Software Engineering (CASE) have demonstrated the impact of user perception about "usefulness" and "ease of use" and various factors that impact these two constructs on the adoption of technology, namely, training and education, support (technical and management). Table 4 summarizes the results of these empirical studies that clearly demonstrate the correlation between the different constructs proposed by the TAM model.

Venkatesh and Davis (2000) conducted a longitudinal study of four different systems at four different organizations to validate TAM2. They measured the model constructs at three points in time at each organization: pre-implementation, one month post-implementation, and three months post implementation. This study determined that subjective norm exerts a "significant direct effect on usage intentions over and above perceived usefulness and perceived ease of use for mandatory (but not voluntary) systems" (Venkatesh and Davis, 2000, p.198). Subjective norm also "significantly influenced perceived usefulness via both internalization, in which people incorporate social influences into their own usefulness perceptions, and identification, in which people use a system to gain status and influence within the work group and thereby improve their job performance" (Venkatesh and Davis, 2000, p.198). As individuals gained direct experience with a system over time, they relied less on social information in forming perceived usefulness but continued to judge a system's usefulness on the basis of

Table 4: TAM Model - Empirical Studies

Factor	Technology	Reference
Ease of Use	CASE	(Chau, 1996)
User Involvement	DSS ES	(Guimaraes et al., 1992; Alavi and Joachimsthaler, 1992) (Yoon et al., 1995; Guimaraes et al. 1996),
Perceived Importance and Usefulness, Impact on Job	CASE DBMS ES	(Chau,1996) (Grover and Teng, 1992) (Yoon et al., 1995; Guimaraes et al., 1996),
Training & Education	DSS ES MRP	(Sanders and Courtney, 1985; Guimaraes et al., 1992; Alavi and Joachimsthaler, 1992) (Kunnathur et al., 1996) (Ang et al., 1995)
Presence of a champion	DSS ES ISDN	(Palvia and Chervany, 1995; Curley and Gremillion, 1983) (Kunnathur et al., 1996) (Lai, 1997)
Management Support	ES DBMS DSS	(Yoon et al., 1995; Guimaraes et al., 1996) (Grover and Teng, 1992; Gordon and Gordon, 1992) (Guimaraes et al., 1992; Sanders and Courtney 1985)

potential status benefits resulting from use. The extended model was strongly supported at all three points of measurement.

Chau and Tam (1997) conducted an extensive study involving senior executives from 89 organizations to determine the factors that impact the adoption of open systems in organizations. Their model is based on the framework developed by Tornatzky and Fleischer (1990) that postulates that the adoption of innovation in an organization is influenced by environmental, technological and organizational context. Their model considered the impact of environmental factors (market uncertainty), technology characteristics (perceived benefits, barriers, importance of standard compliance, interoperablity, and interconnectivity) and organization context (formalization, complexity of infrastructure, and satisfaction with existing systems). Their study determined that there was no significant relationship between market uncertainty and the adoption of open system technology. One possible explanation offered was the fact that this technology impacts the IT infrastructure only and hence external environmental variables have less impact on its adoption. Perceived barriers had a significant impact as opposed to perceived benefits, which had no impact. The key contribution of this study was the inclusion of perceived barriers as an important determinant of the success of the technology. This was not covered by Rogers' diffusion of innovation framework.

In summary, most of the IT implementation studies are either application specific (databases, MRP, EDI, etc) or related to end user computing (Expert Systems, Decision Support Systems, CASE, etc.). Little research has been done to study the implementation of technologies that affect the computing infrastructure of entire organizations. Lai's work on ISDN implementation and Chow and Tam's work on open system adoption are the two exceptions.

CONTEXTUAL FACTOR RESEARCH

Unlike many other technologies that have a specific application, the scope of global distributed infrastructures is company wide and it affects every component of the IT infrastructure. The direct impact of this technology is confined to the IT infrastructure and software applications only and is transparent to the business process it supports. Hence, the impact of the environmental factors on the adoption and implementation of this technology will be minimal (Chau and Tam 1997). Moreover, since the end users of this technology are software applications, the impact of user and task contextual factors will not be as significant as it is when the end users are humans.

The following sections identify the relevant contextual factors from literature that have been empirically shown to impact the successful adoption and implementation of information technology. Though the focus of this research is on organizational and technological factors, the affect of the user, task and environmental factors has also been incorporated in order to develop a broader understanding of the subject. From the literature review, it is observed that various researchers have utilized distinctive variables in assessing implementation, therefore making synthesis difficult. To resolve this, significant factors were abstracted and aggregated into a reduced set of generic categories based on their nature and similarity. These categories are further conceptualized into finer dimensions that are documented in tabular form. The columns in the following tables (Tables 5 to 8) have been made consistent with the grounded theory methodology of identifying categories, sub-categories and properties.

Organizational Factors

Numerous researchers have studied the impact of various organizational factors on the successful implementation of technology. The following key categories, summarized in Table 5, emerged from the literature review:

- organization's alignment of IT goals with its business goals
- organization's understanding of technology
- organization's commitment to technology
- organization's technological competence.

Alignment of the IT goals with the business goals has been identified as a key factor that impacts the success IT systems. It is one of the key challenges and issue facing IS managers (Niederman et al., 1991; Watson and Brancheau, 1991; Moores, 1996). Earl (1993), in his research on Strategic Information Systems Planning (SISP), states that success of these systems is based on continuous integration between the IS function and the organization's goals. Ang et al. (1995) state that clarity in the goals and objectives of the IT organization and as they relate the goals of the organization is critical to the success of the projects. Strategic advantage and operational efficiency have been identified as the key reasons why organizations use distributed systems technology (Chengalur-Smith and Duchessi, 1999). Therefore, one of the organizational factors that will impact the success of distributed middleware infrastructure technology is its use to create operational and strategic advantage for the firm.

However, there is no theoretical framework to determine and quantify the impact and contribution of technology on the firm's operational and strategic goals. This is a key

Table 5: Organizational Contextual Factors

Contextual Factor (Category)	Contextual Factor (Sub Category)	Contextual Factor (Property)	Technology
IT goal alignment with business goals (Niederman et al., 1991; Watson and Brancheau, 1991; Brancheau et al. 1996; Moores, 1996)	Operational efficiency (Chengalur-Smith and Duchessi, 1999)	▪ Lower costs & increase margins ▪ Facilitating and managing business process redesign (Brancheau et al. 1996)	Key Issues in Technology (Brancheau et al., 1996) Strategy (Chengalur-Smith and Duchessi, 1999)
	Integration of the IT function with the organizations' goals (Earl, 1993)	▪ Clear IT goals and objectives (Ang et al., 1995)	MRP (Ang et al., 1995) SISP (Earl, 1993)
	Strategic/competitive advantage (Chengalur-Smith and Duchessi, 1999)	▪ Rapidly deploy products, services, and operations to new emerging market locations around the globe. ▪ Faster time to market for new products & services	Strategy (Chengalur-Smith and Duchessi, 1999)
Organization's understanding of Technology/ Technology Awareness (Lai and Guynes, 1994) (Meyer and Curley, 1991)	Understanding of cost of technology		ISDN (Lai and Guynes, 1994) ES (Meyer and Curley, 1991)
	Understanding of benefits of technology (Yoon, 1995)	▪ Relative advantage(Lai, 1997) ▪ User's understanding of data processing (Magal et al., 1988) ▪ Understanding of user's business problem (Magal et al., 1988)	ISDN (Lai 1997) IC (Magal et al. 1988) DSS (Guimaraes et al. 1992) ES (Yoon 1995)
	Understanding of the limitations of technology		

Contextual Factor (Category)	Contextual Factor (Sub Category)	Contextual Factor (Property)	Technology
Organization's commitment to technology	Top management support (Magal et al., 1988;Guimaraes et al., 1996; Guimaraes et al., 1992; Ang et al., 1995; Lai, 1997)	▪ Organizational acceptance of concept (Magal et al.,1988) ▪ Top management in favor of the concept (Magal et al., 1988) ▪ Presence of a champion (Lai & Guynes, 1994;Palvia and Chervany, 1995) ▪ Belief that time and resources spent on the technology are a wise investment (Guimaraes et al., 1992) ▪ Provide necessary help and resources (Yoon, 1995) ▪ Promote and encourage use of technology(Yoon, 1995; Magal et al.,1988)	IC (Magal et al., 1988) DSS (Guimaraes et al., 1992;Palvia and Chervany, 1995) ES (Guimaraes et al., 1996) MRP(Ang et al., 1995) ISDN (Lai, 1997;Lai and Guynes, 1994)
	Commitment of the end-user organization to the technology (Magal et al., 1988)		IC (Magal et al., 1988)
	Commitment of the IT management to technology	▪ Provide training to the technical staff (Magal et al., 1988) ▪ Well defined career path for the technical staff (Magal et al., 1988)	IC (Magal et al., 1988)
Organizations technical competence	End user organization	Experience with Technology (Magal et al., 1988; Guimaraes et al., 1992) Technological competence of the staff (Ang et al., 1995)	DSS (Guimaraes et al., 1992) IC (Magal et al., 1988) MRP (Ang et al., 1995)
	IT organization	▪ Technical skills (Guimaraes et al., 1996; Yoon,1995)	ES (Guimaraes et al., 1996; Yoon ,1995)
	Top management		

gap in research and one of the aims of this study is to determine the various dimensions for the measurement of operational and strategic impact of these systems and identify the techniques used in the field for measurement and communication of the contribution of this technology.

An Organization's Technology awareness (Lai and Guynes, 1994) or understanding (Earl, 1993) has been consistently identified as a key success factor. This concept has been conceptualized in different ways including understanding the advantages of the technology (Lai, 1997), understanding the business problems and issues that the technology is trying to resolve (Magal et al., 1988), and understanding the time and resources being spend (Guimaraes et al. 1992) on the solution. The literature review has not revealed a comprehensive understanding of this concept and that is another key gap in research.

Since the cost of deployment of middleware technology is high and the entire infrastructure must be in place before it can be effectively used, considerable time and resources have to be deployed before any gains are visible. Hence, an in-depth understanding of the costs, benefits and the limitations of the technology are critical so that the organization can have the insight and confidence to commit the time and resources for a long-term high-risk project.

Organizations that understand technology are more committed to using it for the right reasons and since they are more aware of the risks and limitations, they can take countermeasures to contain the risk.

An organization's commitment to the technology as a key factor for success has been studied extensively across various technologies including databases, expert systems, decision support systems, information centers and others. Magal et al. (1988) identified this as a key concept in the study of information center success and identified some of the key subcategories and properties of this category.

This commitment manifests itself as support of the top management (Guimaraes et al., 1992; Magal et al., 1988;Yoon, 1995), belief that the time and resources spend on the technology are a wise investment (Guimaraes et al., 1992) and an overall favorable view of the technology. Lack of management support and commitment has been shown to lead to failure of systems (Rudelius et al.1982; Lee, 1986). Commitment to the technology is , essential to receive the time and resources needed and also to overcome resistance to changes in work environment created by the new technology (Yoon, 1995).

Organization's technical competence is vital for successful technology implementations. This includes the competence of the application developers and technology providers (Guimaraes et al., 1996; Magal et al., 1988), and experience with technology (Guimaraes et al., 1992).

In summary, though the key organizational contextual factors have been identified, they are not well developed and the researchers have used different dimensions in their research.

Technology Factors

Several researchers have studied the impact of technology factors on the successful implementation of IT. Their models have been primarily derived from Rogers Innovation Diffusion Theory and their key contribution has been operationalization and application of the theory to various technologies like ISDN, Open Systems, Information

Centers, etc. The following key technology factors, summarized in Table 6, have been identified in literature:

- relative advantage
- complexity
- compatibility
- barriers to technology
- quality of service.

Relative Advantage is one of the key reasons why new technology is adopted and implemented in organizations (Rogers, 1983). Relative Advantage can be in the form of new functionality (Lai and Reeh, 1995) or new benefits (Chau and Tam, 2000; Meyer and Curley, 1991). The benefits can be in the form of more flexibility (Henderson & Schilling, 1985), integration, and interoperability with existing systems (Chau and Tam, 1997). Other benefits include better utilization of resources, more choices of hardware and software, standardization, more freedom from the constraints of existing technology (Chau and Tam, 2000), and increased adaptability (Henderson & Schilling, 1985).

Complexity of the technology negatively impacts its adoption and implementation (Rogers, 1983). Complexity makes it difficult to understand, implement, use and manage the technology. Lai and Reeh (1995), in their study of the implementation of ISDN technology, identified the difficulty of understanding, implementing, using and managing technology as the key dimensions of complexity that impact adoption and implementation.

Compatibility of the technology with the existing systems lowers the barrier to entry and promotes easy integration into the existing environment (Rogers, 1983).

Table 6: Technological Contextual Factors

Contextual Factor (Category)	Contextual Factor (Sub Category)	Contextual Factor (Property)	Technology
Relative Advantage (Rogers, 1983)	Technology Functionality (Lai and Reeh, 1995)	▪ Types of Services (Lai and Guynes, 1994) ▪ New Features	ISDN (Lai and Guynes, 1994)
	Benefits of the technology (Chau and Tam, 2000; Meyer and Curley, 1991)	▪ Better utilization of IT resources (Chau and Tam, 2000) ▪ Flexibility (Henderson & Schilling, 1985) ▪ Integration and interoperability(Chau and Tam, 1997) ▪ No longer constrained by proprietary systems (Chau and Tam, 2000) ▪ More choice of hardware and software (Chau and Tam, 2000) ▪ Technology Adaptability (Henderson & Schilling, 1985) ▪ Technology Standardization (Lai, 1997)	Open Systems (Chau and Tam, 2000) DSS (Henderson & Schilling, 1985) ES (Meyer and Curley, 1991)
Technology Complexity (Rogers, 1983)	Technology Manageability (Lai and Reeh, 1995)	▪ Ease of Maintenance (Lai and Reeh,1995) ▪ Ease of Administration (Lai and Reeh, 1995)	ISDN (Lai and Reeh, 1995)
	Difficulty of understanding (Lai and Reeh, 1995)		
	Difficulty of implementing(Lai and Reeh, 1995)		

Contextual Factor (Category)	**Contextual Factor (Sub Category)**	**Contextual Factor (Property)**	**Technology**
	Difficulty of Use(Lai and Reeh, 1995)		
Compatibility (Rogers, 1983)	Integration with existing systems	▪ Hardware compatibility (Lai and Guynes, 1994) ▪ Software compatibility (Lai and Guynes, 1994) ▪ Compatibility with communications environment (Lai and Guynes, 1994)	ISDN (Lai and Guynes, 1994)
Barriers to Technology (Chau and Tam, 1997)	Implementation Gap (Chau, 1996)		
	Satisfaction with existing systems (Chau and Tam,1997)		
	Cost of Technology	▪ Cost effective solutions (Magal et al., 1988) ▪ Cost of Migration (Chau and Tam, 2000)	IC (Magal et al. 1988; Essex et al., 1998) Open Systems (Chau and Tam, 2000)
Output Quality or Quality of Service (Venkatesh and Davis, 2000)	Performance of Service (Lai, 1997) Quality of Support Services (Magal et al.,1988) Quality of Transitional Support (Chau, 1996)	▪ System performance (Magal et al.,1988; Lai,1997) ▪ System Quality (Lucas and Spitler, 2000)	IC (Magal et al.,1988) CASE (Chau 1996) Work Stations (Lucas and Spitler 2000) ISDN (Lai 1997)
	Reliability of Service (Magal et al., 1988; Essex et al.,1998)		IC (Magal et al., 1988; Essex et al. 1998)

Compatibility with the existing hardware, software, and the communications infrastructure has been positively associated with the successful implementation of ISDN technology (Lai & Guynes, 1994).

The key barriers to technology adoption and implementation have been identified in various studies. Implementation Gap has been identified as a barrier that negatively influences the user perception of ease of use of the technology (Chau, 1996). Chau (1996) defines Implementation Gap as the difference between the current state and the new state that will be the result of the implementation of the technology. This includes the gap in knowledge, skills, procedures, and the differences in functionality, service and the other promised benefits of the new technology. The other barriers are the cost of technology with the cost of migration (Chau and Tam, 2000) and the ongoing cost of the solution as the two relevant components. Higher satisfaction with existing systems creates the inertia that negatively impacts the adoption of new systems (Chau and Tam, 1997).

Venkatesh and Davis (2000) identified Quality of Service as one of the key factor that influences the perceived usefulness of the system. Essex et al. (1998) identified this as a key factor in the study of information center success. The following concepts are included this category: Reliability of the systems (Magal et al., 1988), System performance (Magal et al., 1988), quality of support (Magal et al., 1988) including transitional support that is provided during the transition period (Chau, 1996).

Task Factors

The two key task factors, summarized in Table 7, that have been identified are Task Importance and Task Difficulty. These have been extensively studied for technologies that provide tools for complex problem solving like Expert Systems

Table 7: Task Contextual Factors

Contextual Factor (Category)	Contextual Factor (Sub Category)	Contextual Factor (Property)	Technology
Task Importance (Guimaraes et al., 1996)			ES (Guimaraes et al., 1996)
Task Difficulty (Guimaraes et al., 1996; Guimaraes et al., 1992)	▪ Task Interdependence (Guimaraes et al., 1992) or variable interdependence (Yoon, 1995) ▪ Task Structure (Guimaraes et al., 1992;Yoon, 1995) ▪ Task Variety (Guimaraes et al., 1992) ▪ Task Uncertainty (Guimaraes et al., 1992) ▪ Level of expertise needed (Yoon, 1995)		ES (Guimaraes et al., 1996; Yoon, 1995) DSS (Guimaraes et al., 1992)

(Guimaraes et al., 1996) and decision support systems (Guimaraes et al., 1992).

Task importance is described as an activity that has a high payoff to the organization (Hayes-Roth and Jacobstein, 1994; Guimaraes et al., 1996). Successful systems are designed to solve problems that are core to the business.

Task difficulty, as the name suggests, characterizes the level of difficulty of performing a task. Systems that address problems that are perceived by users as being difficult are likely to be perceived as more helpful and successful (Guimaraes et al., 1996). Guimaraes et al. (1996) identified and defined the following key dimensions of task difficulty for decision support systems: interdependence, structure, variety, uncertainty, and level of expertise needed.

Task interdependence, also characterized as variable interdependence (Yoon, 1995), is a measure of the dependence on others for performing a task.

Task Structure is related to the extent to which a decision can be reached by well-defined processes and quantitative methods. Structured decisions can be reached by quantitative procedures and unstructured decisions require heuristics, intuition and experience. Semi-structured decisions have some of the characteristics of both.

Task variety is a measure of how mechanized a task is. It is similar to task routineness or repetitiveness and is defined as the frequency of unexpected novel events that occur in the process of performing the task.

Decision uncertainty is a measure of the level of uncertainty associated with events and resulting outcomes during the performance of the task. Random decisions are the ones in which the events are random and no probabilities can be assigned. Probabilistic decisions are the ones in which events and their associated probabilities are known. Deterministic decisions are the ones where all events and their outcomes can be determined accurately.

The most difficult tasks are those which are highly dependent on multiple participants, have random events that can not be predicted in advance and for which only experience and intuition can provide a decision. The easiest tasks are those that depend on only one person for their accomplishment, are highly structured, have events that can be accurately predicted, and that can be handled by well-defined procedures and techniques. Users perceptions of the benefits of technology tend to be higher when it addresses unstructured and uncertain tasks (Guimaraes et al., 1992; Guimaraes et al., 1996) or when it performs repetitive and monotonous tasks.

User Factors

The three key factors, summarized in Table 8, are impact on user, their perceptions about the technology, and their situational context.

Technology impacts both users' ability to perform their job i.e. their job productivity and also impacts their personal life. Chau (1996) classifies these as Near-term (users subjective belief that using a specific system will increase his/her job performance) and Long-term (users perception that the system will have a long-term impact on their life) consequence for the user.

Technologies that enhance both the job productivity and users' personal life have a higher adoption rate (Yoon, 1995). Technologies that increase the productivity without threatening the users find wide acceptance and those that threaten the career and job prospects of the users face resistance (Guimaraes et al., 1996). For example, one of the

Table 8: User Contextual Factors

Contextual Factor (Category)	Contextual Factor (Sub Category)	Contextual Factor (Property)	Technology
Impact on user	Job Impact of Technology (Guimaraes et al., 1996; Yoon, 1995) Or daily productivity	▪ Near Term consequences (Chau,1996)	CASE (Chau 1996) ES (Guimaraes et al., 1996;Yoon, 1995)
	Personal impact of technology Can also be called "career prospects" (Yoon, 1995)	▪ Long term consequences(Chau, 1996) ▪ Increased ability to change jobs ▪ Better opportunities to do meaningful work ▪ Intangible benefits like increased prestige amongst colleagues	CASE (Chau, 1996) ES (Guimaraes et al., 1996)
Perception (Davis, 1989)	Perceived ease of use (Davis, 1989) Perceived usefulness of technology (Davis, 1989)		CASE (Chau, 1996) ES (Guimaraes et al., 1996;Yoon, 1995)
User Situational variables (Alavi and Joachimsthaler, 1992)	Organizational level of user (Guimaraes et al., 1992)		DSS(Guimaraes et al., 1992)
	Training and Education (Kunnathur et al., 1996) Experience		DSS (Alavi and Joachimsthaler 1992) ES (Kunnathur et al., 1996)

leading causes of failure of expert systems has been the user's apprehension that these systems will replace them (Yoon, 1995).

The impact of user perception of technology, postulated by TAM, has been extensively validated in numerous. These perceptions can be related to ease of use of the technology or about its usefulness in performing their jobs or career prospects. As expected, any perceived negative impact increase user anxiety and creates resistance.

Finally, user's situational context also determines their response to technological change. Their educational background, experience and training impacts their perceptions about technology and also their ability to use technology to increase their job productivity (Alavi and Joachimsthaler 1992; Guimaraes et al., 1992; Kunnathur et al., 1996).

Environmental Factors

The impact of external environment on the adoption of technology has been studied in detail. Market competition and uncertainty has been shown to stimulate rapid adoption of technology (Mansfield et al., 1977). Chengalur-Smith and Duchessi (1999) conclude that market leaders are motivated by technology's ability to increase productivity, reduce costs, and reduce cycle time through improved operations. This impact is limited to technologies that have a direct impact on the business process (Chau and Tam, 1997).

SUMMARY

The previous sections identify the various technology adoption and implementation models and the relevant contextual factors from literature that have been empirically shown to impact the successful adoption and implementation of information technology. Figure 4 summarizes the ones that are relevant for this research.

This analysis has provided a comprehensive list of factors that impact technology implementation in organizations. The review reveals that the most of the factors are relevant across different IT application types, though the extent of applicability is different. It is assumed that some of the success factors related to these diverse system types will also be applicable to Distributed Infrastructure success.

GAPS IN THE RESEARCH

A key gap in prior research is an absence of a comprehensive theoretical framework that can be used to identify the key concepts and issues as well as provide an understanding of the complex interaction of process and context in successful technology implementations. Various theoretical models have been developed to address different dimension of the phenomenon of IT implementation in organizations. None of these models provide a holistic view of the phenomenon and provide a "big picture" that can give researchers and practitioners some perspective so that they can tackle the complex issue of management of global distributed infrastructures in a rational way. This study fills this gap.

In empirical studies, researchers have typically used concepts from various models to develop hybrids, develop formal propositions that posit relationships between various factors, identify quantifiable measures of variables, and attempted to empirically validate them for different technologies like Expert Systems, Decision Support Systems, and MRP. These studies fail to provide a more general understanding of technology

Stages of Adoption & Implementation		Contextual Factors (Kwon and Zmud, 1987)				
Cooper & Zmud Model	**Chengalur-Smith & Duchessi Model**	**Technology**	**Organization**	**User**	**Task**	**Environment**
Initiation	Adoption	• Relative advantage • Complexity • Compatibility • Barriers to technology • Quality of service	• IT alignment with business • Understanding of technology • Commitment to technology • Technological Competence	• Impact on user • Perception • Situational Variables	• Task Difficulty • Task Importance	• Competition • Uncertainty
Adoption						
Adaptation	Implementation					
Acceptance						
Routinization	Post Implementation					
Infusion						

Specific Technology
Examples: ISDN, Databases, Expert Systems, CASE tools, etc.

Figure 4: Summary of Implementation Models and Contextual Factors

implementation that can be used across technologies and particularly to global distributed infrastructures.

In reality, for any implementation to succeed, the process has to be right, the conditions must be optimal and the participants have to take the right actions. If the conditions are not optimal and the process is not conducive to the attainment of the organization goals, the participants need a more comprehensive understanding of the phenomenon to be able to identify the issues and take appropriate actions to ensure success.

The following are some of the other gaps in research that this study attempts to fill:

1. Some of the concepts in the literature have received little or no attention from researchers. For example, the role of vendors as a key stakeholder has received scant attention. Similarly, the concepts of triability and visibility (Rogers 1983; Rogers and Shoemaker 1971) have not been explored in past research. These have significant impact in the implementation of technologies that are relatively new, expensive and have a high inherent risk as they impact the entire computing infrastructure of the firm.
2. None of the studies address the complex issues involved in any global implementation of technology. This is crucial when implementing technology in the modern organizations that span continents and time zones.

This study generates this "understanding" and provides a new theory that explores both the contextual and the process dimensions of the successful implementation of global distributed infrastructures. This study also identifies the key stakeholders, their issues, concerns, the strategies they adopt and actions they take to address their problems.

CHAPTER 3

RESEARCH METHODOLOGY AND DESIGN

This section discusses the research objective, research question, methodology and design, and methods of data collection, analysis and validation.

RESEARCH OBJECTIVE

The primary purpose of this research is to develop an explanatory and descriptive theory of middleware implementation in large global organizations. The objective is to develop an integrated framework that:

- takes into consideration the organizational context such as business drivers, strategies, actions, and constraints of the stakeholders;
- accounts for the technological factors that impact the organization and the stakeholders;
- explains why some middleware technology implementations succeed and others fail;
- clarifies the patterns and relationships that can serve as the basis of future theoretical and empirical research; and
- provides direction for practitioners so that they can take appropriate actions and institute appropriate policies and processes to have more predictability and control over middleware implementations.

This research used grounded theory methodology to investigate the implementation of three key middleware technologies in large global organizations to develop the theoretical framework.

RESEARCH QUESTION

The focus of this research is to understand the key factors and processes that affect the implementation of middleware in large global organizations. As such, the core research question is "What are the critical technological and organizational elements that affect the successful implementation of middleware in large organizations?"

As the study progressed and my understanding of the issues increased, it became apparent that rapid changes in the middleware technology and the incongruence in the strategies and actions of the key organizational stakeholders were the key elements that affected the implementation of middleware in organizations. This understanding spawned the following sub questions:

- Who are the key organizational stakeholders? What are the drivers for their goals and strategies, and how do their actions impact the implementation and adoption of middleware technology?
- What are the key drivers for the rapid evolution of the middleware technology?
- Why do some of the middleware technologies fail and others succeed and how does it impact the organization, the stakeholders and their strategies?

Using the grounded theory methodology, a theory was developed by the author to answer these research questions by investigating the use of three standard middleware technologies in large global organizations.

RESEARCH METHODOLOGY

The research objectives of this project strongly influenced the choice of the research methodology. Grounded theory approach was used here with the objective of developing a descriptive and explanatory theory of middleware technology implementation. This approach has been used effectively in information systems research to study how managers evaluate their decision support systems (Toraskar, 1991), how designers use dialog charts (Calloway and Ariav, 1991), and to understand how CASE technology can be used for organizational change (Orlikowski, 1993) among others. Recent years have seen a surge in studies in IS that use the grounded theory methodology (Adam and Wood, 1999; Galal 2001; Scott, 2000) and it was used for this project for the following two reasons:

First, "the intent of a grounded theory study is to generate or discover a theory, an abstract analytical schema of a phenomenon that relates to a particular situation" (Creswell, 1998, pp. 55-56). This approach is useful here because an extensive review of existing literature, documented in Chapter 2, has revealed that there has been no investigation of middleware technology implementation to date. The use of grounded theory for this project facilitated the generation of a substantive model of middleware implementation that can now be "subjected to further empirical testing because now we know the variables or categories from field-based data" (Creswell, 1998, p. 58).

Second, the complex phenomenon of middleware implementation needs to take into account the dimensions of context, process, and the action/interaction of key stakeholders. Grounded theory has these "specific components: a central phenomenon, causal conditions, strategies, conditions and context, and consequences" (Creswell, 1998, p. 58). It is a good fit with the research objectives of this project.

The grounded theory approach is useful because it takes into consideration the goals and strategies of key players, the organizational processes, and the impact of context, processes and causal and intervening conditions. This aligns the research methodology with the research objectives and the desired outcome of the research project - development of a substantive theory of middleware implementation. The next section discusses the research design.

RESEARCH DESIGN

This section describes the research design including the research context, data collection, analysis, and validation procedures.

Three key principles guided the design of the research. First, the focus on generating a theory that had wider applicability dictated the selection of participants from three different backgrounds - end-user firms, consulting firms and technology industry analyst firms. Second, a theoretical sampling process was used to select sites and participants who could contribute to the emerging theory. Third, constant comparison between data and literature and iteration between data analysis and data collection was used to ensure systematic collection and analysis of data.

Research Context

Theoretical sampling was the primary principle that guided the selection of the sites and participants. This process is theory based and the "data gathering [is] driven by concepts derived from the evolving theory" (Strauss and Corbin, 1998, p. 201). Sites and individuals were selected based on their ability to "contribute to the evolving theory" (Creswell, 1998, p.118).

In the initial stages of the project, the focus of the research was participants from organizations that were end users of middleware technology.

As the research progressed, it became apparent that rapid changes in the middleware technology standards and products was one of the key factors that influenced the implementation of this technology in end user organizations. At this stage, the author included participants who had participated extensively in industry standard consortia that defined the middleware standards. This evolving process is referred to as "theoretically directed data gathering" (Glaser and Strauss, 1967).

Participants from consulting firms that work across industry sectors were included to represent a more diverse participant base in order "to confirm or disconfirm the conditions, both contextual and intervening, under which the model holds" (Creswell, 1998, p.119). This process ensures wider applicability of the findings.

Finally, analysts who track the middleware industry were included to ensure that the "descriptions, interpretation and evaluation and thematic ... are right" (Eisner, 1991, p.112). This methodological process is referred to as consensual validation (Eisner, 1991).

Middleware is an infrastructure technology and its key feature is the ability to provide a common global platform that can be used to integrate various software applications across different hardware platforms, operating systems and programming languages. These characteristics influenced the size and structure of the end user organizations that were selected for this research project.

The following sections provide more details on the characteristics of the organizations and participants in this study.

Organization Profile

Distributed Computing Infrastructures, called Middleware in this project, are typically deployed in very large organizations because they are very expensive to build and operate and only a large complex IT environment can fully utilize the capabilities and benefits of such an infrastructure. Small organizations typically depend on single vendor products for their information processing needs and use off-the-shelf products. For this reason this research project focused on companies that spent over $1 billion on IT in the year 2001.

Since distributed infrastructures are geographically distributed but operate as one logical whole, one centralized team that is geographically distributed typically supports them. The end users of this technology are application development (AD) teams that are responsible for writing and supporting business applications that use this infrastructure. If the centralized technology organization is not responsible for the middleware technology in a firm, then there is a very high probability that it is not being used as an infrastructure but as a component of an application. Since the focus of this research is to explore the implementation of middleware as an infrastructure, the author selected organizations that have a centralized infrastructure organization and where the AD teams are aligned with the lines of businesses.

Since this research examines the implementation of three middleware technologies that influenced the distributed computing infrastructure in contemporary organizations, care was taken to select organizations that had implemented all the three technologies in the last ten years.

Middleware technology is evolving rapidly and only organizations with a real business driver to "keep up with the technology" have the appetite to deploy and use these systems in a meaningful way. Financial services industries use this technology extensively and have the geographical and organizational complexity that utilizes this technology to its full potential. Three firms in the financial services industry were selected because they met the above criteria and because the author has access to senior IT managers in these firms due to his ten years of association with this industry.

To ensure that the findings could be generalized to other industry segments, the study also included managers from large consulting firms that typically work across industries. Care was taken to ensure that the consulting firms had a well-defined middleware practice that spanned multiple industry sectors and firms with IT budget over $1 billion.

Technology Industry analysts track the industry standards, products, vendors and the use of technology at end-user firms. Participants from these groups were included to ensure triangulation across the data provided by the end-user and consulting firms.

A summary of the selection criteria and organization types studied appears in Table 9.

Participant profile

In a grounded theory study, the participants selected "need to be individuals who have taken an action or participated in a process that is central to the grounded theory study" (Creswell, 1998, p. 114). They may not be located at a single site; in fact, if they

Table 9: Profile of Organization Types Studied

Organization Type	Selection Criteria	No. of firms
End User	• Size: Over $1,000,000,000 technology spend in 2001 • Structure: Centralized technology infrastructure and business aligned application development • Technology: has implemented all three middleware technologies	3
Consulting Firm	• Well defined middleware practice that spans multiple industry segments.	2
Industry Analyst	• Tracked middleware technology during the last 10 years	3

are dispersed, then they can provide important contextual information" (Creswell, 1998, pp. 113-114). All the individuals selected for this study had been involved with the three generations of middleware. The participants from the end user organizations had been involved with middleware either in an infrastructure or an application development manager role. The participants from consulting firms were partners or senior managers who had been consulting in the middleware specialization of their firms. The industry analysts selected had been tracking the middleware industry for over seven years.

Because the implementation of middleware infrastructures entails considerable expense and risk, the approval and management of these projects typically resides with senior managers. For this research, the author selected only senior managers at the end-user firms who had executive responsibility for the middleware infrastructure or application development projects. From consulting firms, only partners or senior managers were included in this study because they typically work with senior executive managers at the end-user firms. Table 10 summarizes the participant profile.

The titles in the table have been rationalized to indicate the responsibility and the authority vested in the position. For example, the title "managing director" in financial firms selected in this research reflects executive responsibilities over either application

Table 10: Profile of Participants in this Study

Participant Number	Organization	Position	Participant Code	AD Management	Infrastructure Management	Participation in Standards Organization	Comments
1	FS Firm 1	MD	FS1MD1	√	√	√	Chair person of the DCE Standards committee for one year
2	FS Firm 1	MD	FS1MD2	√	√		
3	FS Firm 1	MD	FS1MD3		√		
4	FS Firm 1	VP	FS1VP1	√	√	√	
5	FS Firm 1	VP	FS1VP2		√		
6	FS Firm 1	VP	FS1VP3	√			
7	FS Firm 1	VP	FS1VP4	√			
8	FS Firm 2	MD	FS2MD1	√			
9	FS Firm 2	MD	FS2MD2		√	√	
10	FS Firm 2	VP	FS2VP1	√			
11	FS Firm 2	VP	FS2VP2		√	√	
12	FS Firm 3	MD	FS3MD1	√	√	√	Chair person of the DCE Standards committee for one year
13	CS Firm 1	Partner	CS1PT1	√	√	√	
14	CS Firm 1	Associate Partner	CS1AP1	√	√		
15	CS Firm 2	Partner	CS2PT1	√	√		
16	CS Firm 2	Associate Partner	CS2AP2	√	√	√	
17	CS Firm 2	Associate Partner	CS3AP1	√	√		
18	IA Firm 1	Senior Analyst	IA1SA1				
19	IA Firm 2	Senior Analyst	IA2SA1				
20	IA Firm 3	Senior Analyst	IA3SA1				

Legend: FS = Financial Services, CS = Consulting Services, IA= Industry Analyst, MD= Managing director, VP= Vice President development for a line of business or an entire infrastructure area like middleware. Vice Presidents have accountability at a project level and are typically responsible for critical and high value projects. Similarly, partners have ownership rights in the firm. Associate partners are one level below the partners in consulting firms. Since partners and associate partners at consulting firms are involved in both the activities, they are not categorized as application development or infrastructure specialists. Senior analysts in Analyst firms are responsible for covering a particular area of technology.

Data Collection

Though many activities and types of information contributed to my understanding of the middleware implementation process, four basic types of data sources were used for this research.

First, personal unstructured and semi-structured interviews with the participants provided valuable understanding of the contextual and processual factors. The interviews with the end-users focused on topics of context, stakeholders, their goals and strategies, drivers for technology adoption and implementation, and factors that impacted the fate of the technology at their firms. The interviews with the participants from consulting firms were focused on generalizing the findings to seek broader industry trends related to reasons for adoption and implementation of middleware technology, goals and strategies of stakeholders in different end-user industry segments, and technology trends that impacted the middleware implementation. The interviews with the industry analysts were focused on technology standardization trends, verification of the technology penetration and uptake in end-user companies, their understanding of the factors that affected the broad adoption of middleware technology and their predictions of the future.

Second, some of the participant from the end-user organizations and the consulting firms shared with me the formal presentation files (PowerPoint) that they had made to their senior management, their clients and end-users. For end-user organizations, these presentations are typically made to senior business managers to seek funding, update them on the technology strategy of the firm or to provide them with justifications for the current projects. These presentations are typically prepared over the course of a few weeks and are reviewed by several senior IT managers before they are presented to the final audience. These presentations provided a precise view of the goals, issues, strategies and action plans of the senior managers of these firms. They also provide corroboration with the interview data.

Consulting firms also used presentations (PowerPoint) to pitch middleware technology to their clients. These presentations included their analysis of their client's need, proposed strategy and action plan, and technology recommendation to address the client's needs. These presentations provided valuable insight into the needs of end-user organizations from different industry sectors and were extremely useful in generalizing concepts.

In additions, numerous presentations by middleware vendors were also provided to me by the participants. These presentations provided valuable information about vendor strategies, their view of the benefits of their products, and their recommendations to the users. A total of nine hundred and thirty pages of PowerPoint slides were reviewed for this research project and one hundred and thirty four of these were included in the analysis based on their relevance to this project.

Third, some of the participants made their internal documents available to me. These were primarily white papers written by technologists at end-user firms or by specialists at consulting firms that detailed the middleware solutions, technologies, and recommended architecture and technology strategy. These provided the necessary triangulation between the views of the senior managers and the lower layers of management.

Fourth, over five hundred middleware related industry analyst reports and research bulletins from three leading industry analyst firms, that were part of the sample, were reviewed for this research project. These reports covered middleware industry related topics from January, 1993 to October, 2003. Data from one hundred and two reports was included in this analysis based on their relevance to this project.

In addition, detailed information was collected on company size, structure, corporate strategy (detailed in PowerPoint presentations by CEOs of the firms in analyst forums), IT strategy (detailed in PowerPoint presentations by Chief Technology Officer's of the firms in industry technology forums) and technology spending. These presentations provided a wealth of invaluable information.

The participants were briefed on the scope and intent of the interviews during the process of scheduling these sessions with them .The interview was started with a warm-up opening statement that set the context for the participant and allowed them to ask questions and seek clarifications. The following is an example of a typical warm-up questions used with technology managers:

> In the last 13 years, we have seen two generations of middleware, DCE and CORBA come and go. Now we have the 3rd generation, Web Services, which is in the process of being standardized. We also have other technologies that have evolved in this area - .Net is on the way and Java has evolved from a

> programming language to an infrastructure. You have been in the middle of this all from the very beginning and have done a great job of steering the infrastructure/application development teams through all the gut wrenching changes. In this study, I'm trying to capture your experiences and distill it to come up with something that can be used as a guide to handle similar situations going forward.

This was followed by a broad all-encompassing question that was aimed to elicit maximum coverage of the topic. The following is an example of such a question used with technology managers:

> Can you describe what really happened in your organization with middleware technology? We can start with how and why you selected these (DCE, CORBA and all others you have used), how you went about deploying them, and what really happened when you started using them. We can treat it almost like a post-mortem to get a feel of what we did, why we did it, and what was the result. Also, can you describe your strategy for Web Services?

Then, as they described their experience, the author asked specific questions depending upon what he was seeking in that interview. In the beginning, these were based on the findings of the literature review. During the later stages, these were driven by the data analysis that was being conducted in parallel. The following are some typical questions that were used with technology managers:

> What are the key industry trends that are impacting your IT organization?
> What is your involvement with the vendor consortiums?
> What are you doing about Web Services in your organization?

Appendix C describes in detail the interview guide that was used for this study. Appendix D shows a sample of a complete interview transcript.

Each first interview session ranged in duration from one to two hours. Out of the twenty interviews, fifteen were recorded and transcribed. Five participants denied the request to record the interview.

Follow-up interviews were done with four participants who were deemed to be the most experienced in this area, two in end user companies and two in consulting firms, to discuss the findings and to seek their opinion on the validity of the interpretations. The findings were updated to reflect their input.

Each interview was analyzed and coded before the next one. This process of interviewing was continued to explore and refine the emerging concepts. The process was stopped when all the data could be accounted for by the concepts that had been discovered and refined by this iterative process. This condition is referred to as theoretical saturation. Grounded theory methodology recommends theoretical saturation as a condition to end the process of data gathering (Strauss and Corbin, 1998). Next section describes the data analysis process in more details.

Data Analysis

The methods and procedures of grounded theory described by Strauss and Corbin (1998) were followed to generate categories, their properties, relationships and to weave them into a cohesive theoretical model of middleware implementation that is grounded in empirical data.

A qualitative data analysis software package called ATLAS.ti was used to facilitate the task of data management and analysis in this project (see Appendix E for details). This package is based on the principles of grounded theory and automates the task of textual and graphical analysis by providing convenient and easy procedures for generating codes and writing memos. It also provides facilities for linking these codes to create categories, sub-categories and properties. These categories can then be linked into conceptual networks with relationships that the researcher can define. This tool also provides facilities for creating codes for graphics and pictures. This feature proved to be invaluable as most of the PowerPoint slides were scanned using a scanner and were available only as images for this analysis.

All the interview transcripts, industry analyst reports and images of PowerPoint slides were loaded into Atlas.ti as the data became available over time.

Line by line analysis of each data item was done to develop as many codes as possible and to ensure full theoretical coverage (Hutchinson, 1988). Table 11 shows a sample of quotations and the associated codes. All the PowerPoint slides were analyzed in a similar fashion.

This is referred to as open coding (Strauss and Corbin, 1998). This generated a list of codes that were revised and updated with each subsequent interview and new set of data.

The codes generated by open coding were then organized by themes involving conditions, context, action/interactions, strategies and consequences" (Strauss and Corbin, 1998, pp. 123-142). This is called axial coding. Figure 5, generated from Atlas.ti, shows an example of a category developed by this process of linking various codes developed by axial coding.

Selective coding was then used to relate the categories into a cohesive model. It is defined as "the process of selecting the core category, systematically relating it to other categories, validating those relationships, and filling in categories that need further refinement and development" (Strauss and Corbin, 1998, p. 116).

The iteration between open, axial, and selective coding, and data continued until enough categories and associated concepts had been defined to explain the observations.

Table 11: Sample Quotations and Associated Codes

Sample Quotations from Interviews	Associated Open Codes
"If we start off with DCE and CORBA, the reason why they didn't succeed was that because the industry was moving too fast. It was putting the horse before the cart. Specifically with respect to really getting people more used to describing the interfaces of the programs they were trying to connect via DCE. While there was IDL, I don't think people had in their DNA to actually develop some sort of contract programming style where you describe your interfaces."	Developer acceptance
"I think the industry went wrong in that the vendors, such as Sybase, did not get behind [DCE] all the way and did not provide enough support for the technology. So, while the solution was promised, the fact was that many of the vendors on whom we relied for core technology were not really in the game."	Compatibility with existing technology in use Availability of Complementary products
"Web services have much more richness in describing the interfaces. It also has richness with respect to a lot of the underlying standards it depends on, i.e. the Internet standards. And therefore, unlike previously, when OSF… and OMG were just pushing [their technology and expecting] everybody to join them whether they like it or not; this time it has taken a longer process but there has been much more buy-in from the major vendors around all these technical pieces…[This is] because we have leveraged the Internet standards process. That's why the Web Services has a much better chance of success."	Acceptance of standards by major vendors Technology Standardization Process

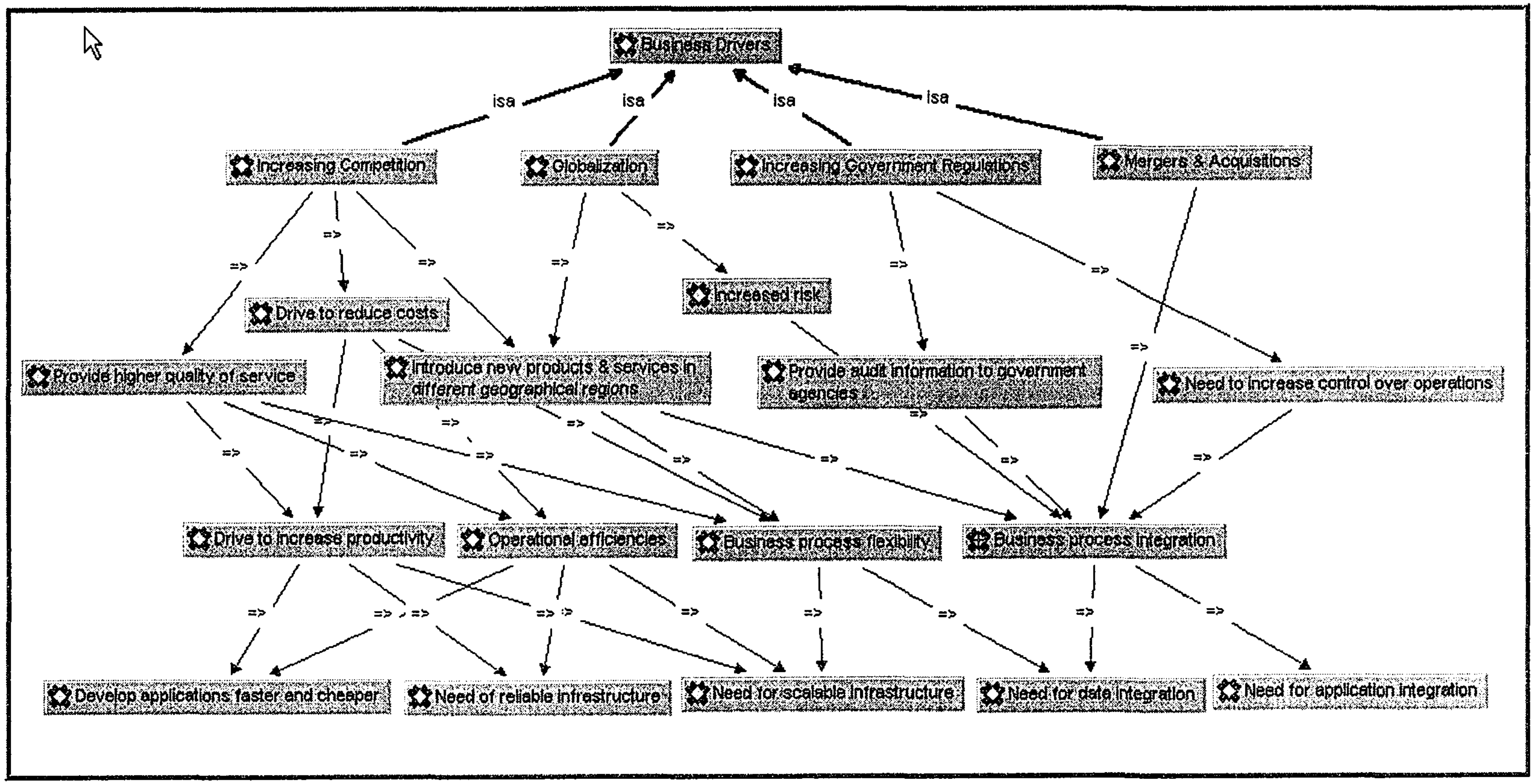

Figure 5: Example of Category Developed by Axial Coding

By the end of the analysis, a total of sixty eight codes were generated in the open coding process. Axial coding resulted in the organization of these codes into thirteen categories. Out of these thirteen, three were deemed to be core categories and the other ten were mapped as sub-categories to the core categories. Table 12 shows the categories and the corresponding sub-categories.

Table 12: List of Categories

Core Categories	Sub-categories
Organizational Context	▪ Business Drivers for Technology ▪ Firm's Technology Environment ▪ Firm's IS Issues
Middleware Implementation Process	▪ Middleware Technology Strategy ▪ Middleware Implementation ▪ Middleware in Use ▪ Consequences
Technological Context	▪ Technology Standardization Process ▪ Acceptance of Standards by Major Vendors ▪ Availability of Complementary Products

Atlas.ti provides facilities for creating abstract relationships between codes and categories, and between categories. Several abstract networks of these relationships were constructed using techniques of axial coding and these are described in detail in Table 13. These were then woven into a theoretical framework by the selective coding process. Figure 6, generated from Atlas.ti, shows the final theoretical framework that emerged from this analysis.

This final model developed is empirically valid as it can account for unique data from different participants and other sources. The model is detailed in Chapter 4 along with detailed descriptions of the concepts and categories.

Appendix F shows all the codes, categories and their relationships.

Table 13: Conceptual Network Descriptions

Network Name & Number	Description
Middleware Implementation Framework Core Categories. Network number I	This network shows the relationship between the three core categories. Networks II, III and IV show more details of these three categories and the linkages between them. Network V shows all the thirteen categories and their relationships in one diagram. See Figure 27 on page 195.
Organizational Context. Network number II	This network describes the relationship between the Organizational context core category and its sub-categories. See Figure 28 on page 196.
Technological Context Network number III	This network describes the relationship between the Technological context core category and its sub-categories. This also provides details of the other categories that make up these sub-categories. See Figure 29 on page 197.
Middleware Implementation Process. Network number IV	This network describes the relationship between the Middleware Implementation Process core category and its sub-categories. See Figure 30 on page 198.
Detailed Middleware Implementation Framework. Network number V	This network shows in detail the linkages between the three core categories and the ten sub-categories. This network is the basis of the model described in Chapter 4. See Figure 31 on page 199.
Business Drivers. Network number VI	This network describes in detail the constituent categories that constitute the Business Drivers sub-category. See Figure 32 on page 200.
Firm's Technology Environment. Network number VII	This network describes in detail the constituent categories that constitute the Firm's Technology Environment sub-category. See Figure 33 on page 201.
Firm's IS Issues Network number VIII	This network describes in detail the constituent categories that constitute the Firm's IS Issues sub-category. See Figure 34 on page 202.
Middleware Technology Strategy Network number IX	This network describes in detail the constituent categories that constitute the Middleware Technology Strategy sub-category. See Figure 35 on page 203.
Middleware Implementation Network number X	This network describes in detail the constituent categories that constitute the Middleware Implementation sub-category. See Figure 36 on page 204.
Middleware in Use Network number XI	This network describes in detail the constituent categories that constitute the Middleware in Use sub-category. See Figure 37 on page 205.

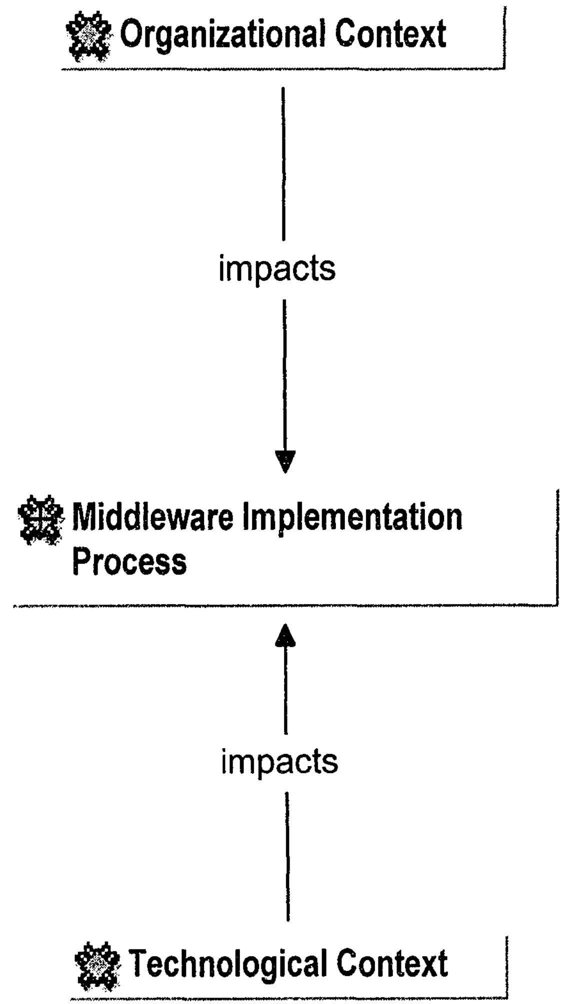

Figure 6: Middleware Implementation Framework Core Categories

Validating the Research

This research used triangulation and member checks to validate the findings of the study.

"In triangulation, the researcher makes use of multiple and different sources, methods, investigators, and theories to provide corroborative evidence" (Creswell, 1998, p. 202). In this project, data obtained from interviews was verified against the PowerPoint presentations provided by the executives, consultants and analysts. This ensured triangulation at the data level. Comparing the views of end-users, consulting firms and industry analysts corroborated the emerging concepts, categories and themes.

In member checks, "the researcher solicits informants' views of the credibility of the findings and interpretations" (Creswell, 1998, p. 202). Follow-up interviews were done with four participants who were deemed by the author to be the most experienced in this area to discuss the findings and to seek their opinion on the validity of the interpretations. The findings were updated to reflect their input. "This technique is considered by Lincoln and Guba (1985) to be 'the most critical technique for establishing credibility' " (Creswell, 1998, p. 202).

CHAPTER 4

RESEARCH RESULTS

This chapter describes the empirical findings of this research and presents the key categories and the resultant theoretical framework that emerged in this grounded theory study.

This chapter is structured to discuss the findings in alignment with the framework of the grounded theory methodology. First, the middleware implementation framework that emerged from this study is discussed in terms of key themes, concepts and related categories, as well as the linkages among them. This is in accordance with May (1986) who recommends that that "in strict terms, the findings [of grounded theory research] are the theory itself, i.e., a set of concepts and propositions which link them" (p. 148). The next section provides details of the data associated with the categories discovered in this research for each of the three technologies, DCE, CORBA and Web Services. Since the data for the organizational context category was similar for all the three technologies, it is discussed first. It is followed by a discussion of the details that set the three technologies apart. The last section summarizes the findings which will be the focus of Chapter 5, a discussion with implications of these findings.

A similar research findings framework structure has been used by other authors and is a common practice in grounded theory based research (Orlikowski 1993; Morrow and Smith 1995).

FRAMEWORK FOR MIDDLEWARE IMPLEMENTATION

The findings from this research show that since 1988, the middleware industry has seen three successive generations of technology standards:

- DCE
- CORBA
- Web Services.

DCE and CORBA did not become dominant in the industry and were rejected by both the vendor and end-user communities. Web Services is an emerging standard and appears to have broad support from both vendors and end users.

Three key themes emerged from the data collected for this study. These themes recurred in each of the three middleware technologies studied.

The first theme concerns organizational factors. Organizational concerns have driven the adoption of successive generations of middleware technologies and appear not to have changed significantly since the early 1990s. Concerns include the firm's business needs, the complexity of their technology environment and the IT issues faced by the technology managers. The broad adoption of the Internet has amplified these concerns and focused greater attention on IT integration with business partners and clients.

This first theme is conceptualized in a core category labeled Organization Context. Organizational Context in this research refers to the organizational conditions and characteristics that create the need for middleware technology as well as impact its implementation, use, and extent of success in the organization.

Three sub-categories, described in Table 14, emerged from the data found in this first, Organizational Context, core category:

- Business Drivers for Technology
- Firm's Technology Environment
- Firm's IS Issues.

These are the internal forces that drive the adoption, implementation and use of middleware technology in firms.

This Organizational Context theme is referred to as Theme I in the rest of the document.

The second theme concerns the middleware implementation process. It refers to a firms' experience with the deployment and use of successive generations of middleware technology and how this evolution shaped an organization's strategy towards the use of next generation of standards.

The data indicates that the adoption of DCE by end user organizations was a strategic enterprise-wide decision made by senior IT managers. One of the key lessons learned with the failure of the use of DCE technology as an infrastructure was that centralized middleware infrastructure was not a feasible solution.

The failure of DCE shaped an organization's strategy towards the adoption and use of the CORBA standard, which was not deployed as an infrastructure technology. Instead, it was adopted by application development teams as another component of the application environment. It helped them write object-oriented code in an organized fashion. The subsequent failure to adopt CORBA by end-users and vendors, however, has resulted in a more cautious technology community.

Table 14: Middleware Implementation - Organizational Context

Category	Sub-category	Description
Organizational Context	Business Drivers for Technology	"Business drivers for technology" are those business imperatives and conditions that make demands on the IT organization to change and evolve in a particular way. This category attempts to capture data that answers the question: "what are the business reasons for adopting middleware?"
	Firm's Technology Environment	"Firm's Technology Environment" refers to the hardware, software, application and data environment of the firm that influence the adoption and implementation of middleware technology.
	Firm's IS issues	"Firm's IS issues" are the problems and issues that the technology managers face in large global organizations. These are typically the driver of a search for solutions that are eventually deployed and used at the firm.

So with Web Services, both end-users and vendors are taking a more pragmatic and restrained approach. Web Services standards are still evolving and being debated in industry consortiums. End-users, instead of rushing-in to take advantage of the latest technology, are addressing fundamental data quality and application architecture issues.

This second theme is conceptualized in a core category labeled Middleware Implementation Process. Implementation process refers to the actions and interactions of a firm's managers in response to day-to-day technology issues. This is influenced by the technological and organizational context that shape the issues they face, the strategies they adopt and the ability of the technology to meet their needs. The following four sub-categories, described in Table 15, emerged from the data found in this second, Middleware Implementation Process, core category:

- Middleware Technology Strategy
- Middleware Implementation
- Middleware in Use
- Consequences.

This Middleware Implementation Process theme is referred to as Theme II in the rest of this document.

The third theme concerns the technology context. The data suggests that the previous generations of middleware technology standards have not been able to meet the needs of end-users.

The technology standardization process that led to the development of the two previous standards, DCE and CORBA, was not optimal. The process lacked the rigorous evaluation by the broad technology community, including individual contributors and

Table 15: Middleware Implementation Process

Category	Sub-category	Description
Middleware Implementation Process	Middleware Technology Strategy	This sub-category refers to stakeholders' assessment about the nature and direction of computer-based technologies, reasons for acquiring and implementing middleware technology in relation to a firm's objectives and goals, and likely economic and non-economic benefits to the firm. "It includes their understanding of the motivation or vision behind the adoption decision and its likely value to the organization" (Orlikowski and Gash, 1994, p.183). This category captures the data that answers the questions: "What should the technology organization be doing? Why they should be doing it? What are the ends and what are the benefits?"
	Middleware Implementation	This sub-category refers to the coordinated set of tactics used by stakeholders to implement the middleware technology strategy and includes their deliberate and planned actions-interactions in relation to other organizational stakeholders, vendors, and standard creation organizations. It also includes how the technology was deployed in terms of geographical, organizational and technological scope. This sub-category captures the data that answers the question: "what did managers do to implement their middleware technology strategy?"
	Middleware in Use	This sub-category conceptualizes the conditions associated with the use of the technology deployed and captures the factors that enhance success or contribute to failure. For middleware technology the following dimensions were found to be relevant: ▪ Compatibility with the firms business environment ▪ Compatibility with technologies already in use at the firm ▪ Technology complexity ▪ Developer acceptance. This sub-category captures the data that answers the question: "what are the factors that 'make-or-break' a technology once it is deployed at the firm and is available for use."
	Consequences	This sub-category captures the complex technological and organizational impacts arising from the implementation of middleware technology in the organizations.

industry experts, who have traditionally led the development of extremely successful Internet standards. As a result, these earlier standard were developed in isolation and failed to meet real-world requirements.

Both DCE and CORBA standards did not have the support of some key platform vendors including Microsoft and Sun Microsystems. Both the vendors produced their own competing products and standards and this negatively impacted the widespread adoption of the standards.

This also impacted the availability of key complementary products like databases because the producers of these products did not support these standards. Typically, these vendors produce products tailored to the broad market needs and they did not see any benefit to support DCE and CORBA standards.

This third theme is conceptualized in a core category labeled Technology Context. Technology Context, in this research, refers to those characteristics of middleware technology that influenced the strategies that managers adopted to address the issues they faced, or that affected the implementation, use and extent of success the technology had in the organization. The following three sub-categories, describe in Table 16, emerged from the data found in this third, Technological Context, core category:

- Technology Standardization Process
- Acceptance of Standards by Major Vendors
- Availability of Complementary Products.

These are the forces, external to the organization, that impact the adoption, implementation and success of middleware technology in firms.

Table 16: Middleware Implementation - Technology Context

Category	Sub-category	Description
Technology Context	Technology Standardization Process	This sub-category refers to the process by which the middleware technology standards were created and their ability to meet the needs of users. The following dimensions are relevant: ▪ expertise and knowledge of the participants ▪ openness of the process so that all participants, including individuals and organizations, are free to contribute ▪ well-defined approval process to ensure that the standards have been reviewed and critiqued by experts ▪ ability of the standards to meet the needs of end-users ▪ demonstrable working standards that could be developed into IT products. This sub-category captures the data that answers the question: "were the middleware standards created with an open process so that experts and innovators in the field participated to ensure that the standards meet the needs of end-users?"
	Acceptance of Standards by Major Vendors	This sub-category refers to the support of standards by major vendors who dominate the IT technology market. This research discovered that support by the major vendors is a key factor that leads to the success of standards in the middleware market.
	Availability of Complementary Products	This sub-category refers to the availability of middleware products that conform to the standards. This also includes the availability of complementary products that integrate with the middleware technology. This sub-category captures the data that answers the question "did the broad vendor community accept the standards and develop products that meet the requirements of the end user community?"

This Technological Context theme is referred to as Theme III in the rest of this document.

The findings indicate that the implementation of middleware technologies in large firms can be best understood by taking into consideration two complementary and interactive phenomena:

- From an organizational perspective both business drivers and characteristics of the firm create the need for middleware technology and influence its appropriation and use. The ability of each successive generation of middleware standards to meet the needs of the firm influence and shape the approach and strategy for the use of the next generation of technology.
- From a technological perspective, the changes in middleware technology standards and products, their ability to meet an organization's needs, and the extent of their adoption by vendor and user communities shape the strategies of key players and their use of middleware technology.

These two phenomenon recurred in accounts of the observations and interpretations participants made about their experiences with middleware technologies. They capture the two ends of the framework given in Figure 7, which shows the linkages among the three core categories that emerged from this research. The framework offered also captures the dynamic and evolving process of middleware implementation as it unfolds within the organizational and technological context.

This framework accounts for the variations of the use and associated consequences of the implementation of the three successive generations of key

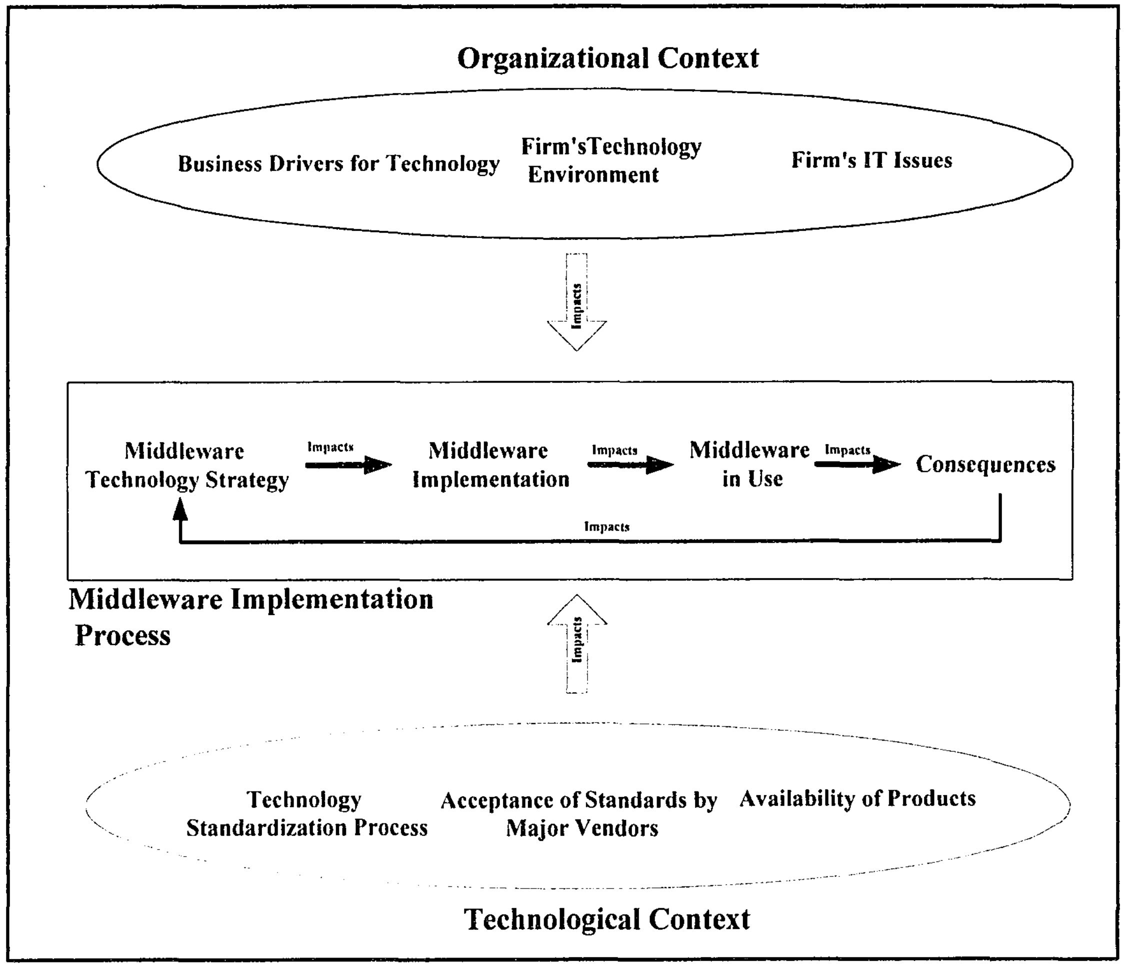

Figure 7: Middleware Implementation Framework

middleware technologies in large global firms. The data discussed in the following sections substantiates this framework.

These findings are derived from the data collected from technology managers, consultants and industry analysts who experienced deploying, using and analyzing the three generations of middleware technologies, DCE, CORBA and Web Services. Even though the participants discussed all the three generations of technology in the same interview, this section logically separates the findings by technology to highlight relevant characteristics and consequences.

The data for each technology, DCE, CORBA and Web Services, is organized to correspond to the three core categories discussed previously, which are:

- Organizational Context for Middleware Implementation
- Middleware Implementation Process
- Technology Context for Middleware Implementation.

The core categories and sub-categories have been described in more detail in Tables 14, 15, and 16.

The data shows that the middleware implementation framework depicted in Figure 7 is empirically valid as it can account for unique data for each middleware technology, as well as for a generalized pattern across the three technologies studied (Eisenhardt 1989).

Since the findings show that the organizational context is common for all the three technologies, it is discussed first to avoid repetition. This is followed by the implementation process and technological context details which are unique for each of the technologies studied.

ORGANIZATION CONTEXT FOR MIDDLEWARE IMPLEMENTATION

The data in this section corresponds to Theme I discussed previously. The various categories that provide more details are described in Table 14. This category describes the forces internal to the organization that drive the implementation of middleware technology in firms.

Some of the participants who had to make the business case for the adoption and use of expensive middleware technology focused on organizational forces that drove the adoption and implementation decision of middleware. These participants made their formal executive presentations available to the researcher. Material from these presentations, along with those from vendors and consulting firms who had to make similar business case for middleware, therefore, is offered in the following sections and formed the raw material for deriving much of the findings for this research.

As identified in Figure 7, this research finds that the following three factors influenced the adoption and implementation of middleware in large organizations:

- Business Drivers for Technology
- Firm's Technology Environment
- Firm's IS Issues.

These factors are described in detail in Table 14. The participants indicated that these organizational forces have not changed significantly over the last decade. In the early 1990s, technology managers deployed DCE and CORBA in response to the same challenges that is now driving the adoption of Web Services. As FS1MD1 explained:

> If you were to write the requirements document for DCE and compare it to what we need now, you'd say that you can still map those requirements [to those for Web Services] and they are [still] not being met.

The following sections provide more details on these factors.

Business drivers for middleware technology

Figure 8 shows a sample of the data provided by the participants that depicts the key business drivers for middleware use. The following are some of the key trends identified from the data:

- Mergers and acquisitions create the need to integrate business processes to gain synergy and reduce costs.
- Globalization drives the need to introduce new products and services in different geographical locations and manage increased risk.
- Governmental Regulations drive the need to improve controls over a firm's business operations and provide audit information to governmental agencies.
- Competition forces the firm to introduce new products and services, reduce costs, increase productivity, and provide higher quality of service to customers.

These trends have resulted in a reduced shelf life for software applications and forced the IT organizations to look for ways to develop software applications faster and cheaper. These applications need to be more scalable and reliable to handle increasing loads and higher customer expectations. The need for business process flexibility and integration is driving the need for application and data integration.

These trends have put pressure on IT managers to provide technology to enable integration and more flexible, scaleable and reliable business processing that also provide strategic advantage and increased efficiencies.

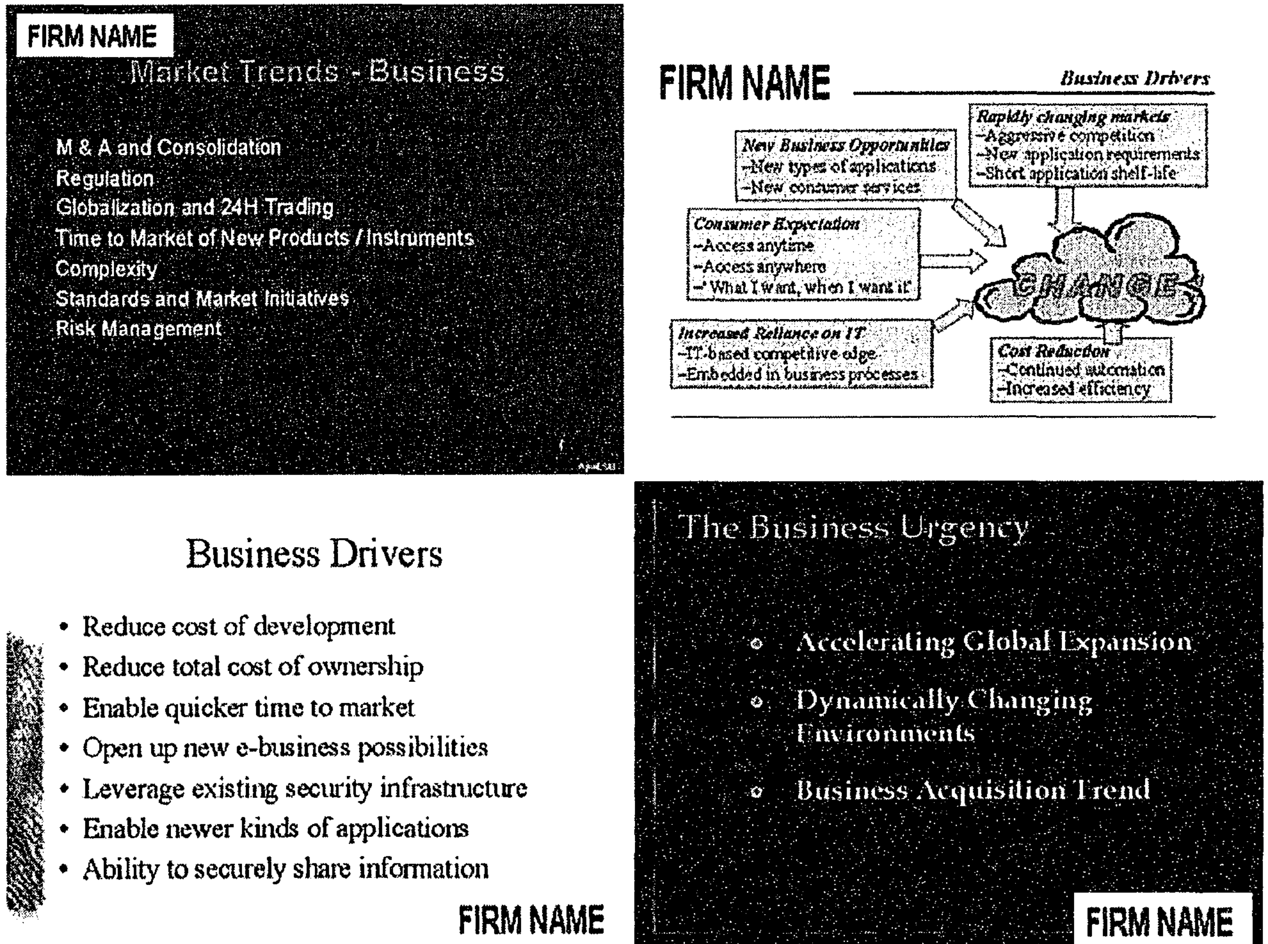

Figure 8: Middleware Business Drivers

Source: Data provided by the research participants

These IT imperatives, coupled with a firm's technology environment, which are discussed in the next section, created the issues that drove the need for middleware at most firms

Technology Environment of the Firm

All the firms studied have a complex technology environment. These firms have multiple networks, operating systems, and hardware and software development platforms.

Just to give an idea of the scale of the complexity of the technology environment, FSFA has over 8,500 applications with over 10,000 interfaces deployed over 15,000 servers listed in their application tracking system. These applications are spread across multiple hardware, operating system and programming platforms. AS FS1MD1 commented:

> Look at App Track [firm's application tracking system - name changed to preserve the identity of the firm]. We have thousands of applications. And you can look at the interfaces they have. Lots of them have tens of interfaces. And these are through to other "islands of technology" that use a different technology.

FS1 uses 235 categories of technology products that include over 1400 distinct products from over 600 vendors. Figure 9 shows sample data from the participants that provides a glimpse into the complex technology environment in typical large corporations.

The interviews with the consulting firms confirmed that most large organizations have a similar environment. As CS1PT1 indicated:

> They [large organizations] are all same. Lot of it is history. Technology evolves and you keep adding [hardware and software] to your [technology] environment... Other stuff you inherit when you buy other companies.

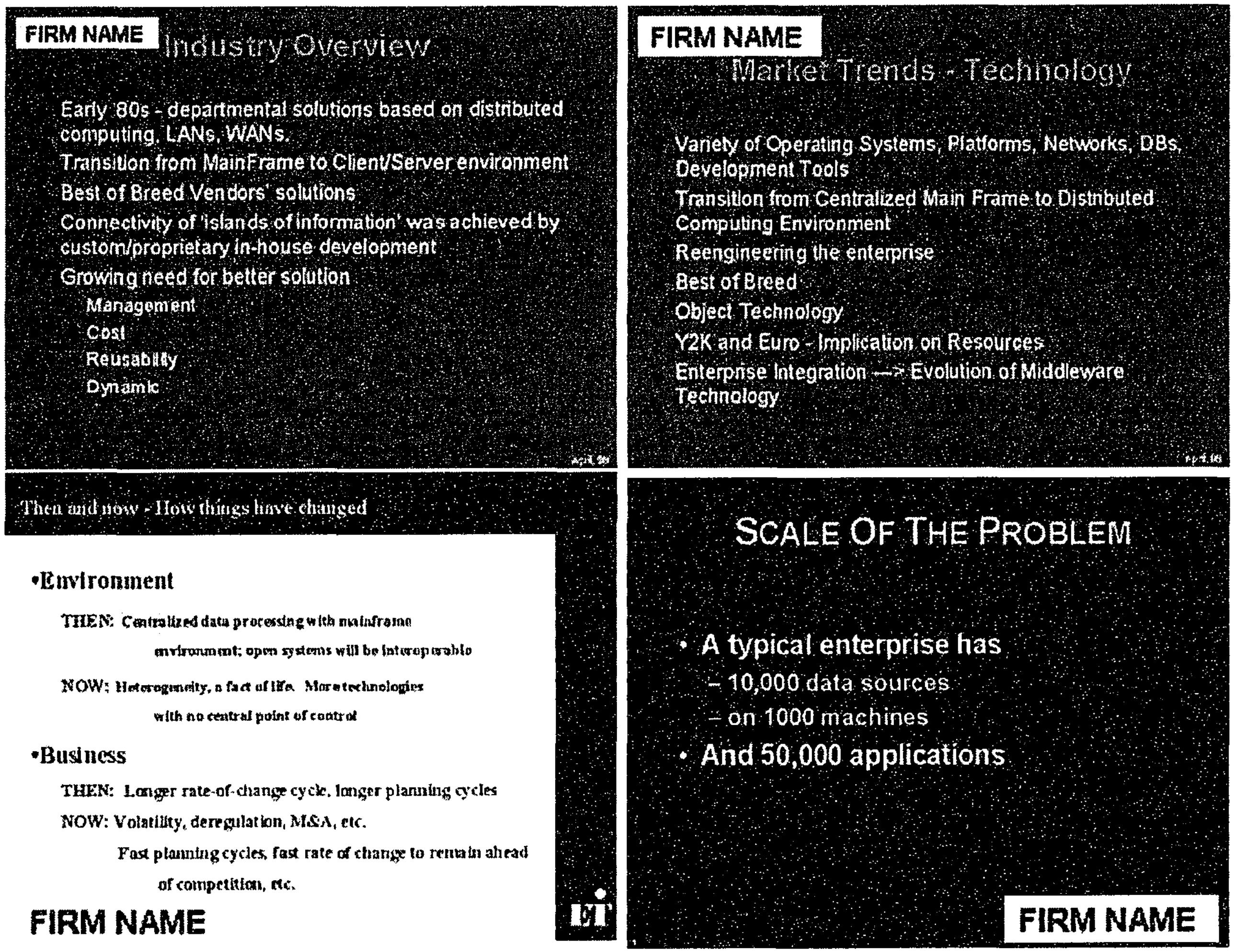

Figure 9: Firms Technology Environment

Source: Data provided by research participants

> Reality is that no one technology can meet all the needs of the [different lines of business in a firm], so they all use whatever suits them the best.

Large firms have a complex technology environment due to technology evolution, mergers and acquisitions and because one technology cannot meet the complex business needs of a global organization.

This complexity of the technology environment, coupled with the business drivers, creates the IS issues that drive the adoption and use of middleware technology. The next section documents the key IS issues identified by the participants in this research.

Firm's IS Issues

Sample data in Figure 10 shows that the infrastructure managers recognize that there are signification IS issues in the firm's distributed environment.

The disparate technology infrastructure and the natural evolution of applications in response to demands from a firm's businesses have resulted in a heterogeneous application environment. This naturally leads to increased complexity, costs and management problems.

Specifically, the participants identified application and data integration as the key IS issues that drive the need for middleware technology.

For example, a typical large enterprise has thousands of data sources, referred to as "distributed islands of dissimilar data" by CS2AP2. These data sources are used for a large number of applications that run on diverse systems. There is a business imperative to keep data semantically consistent and this requires that applications interoperate to ensure data sharing and data integrity.

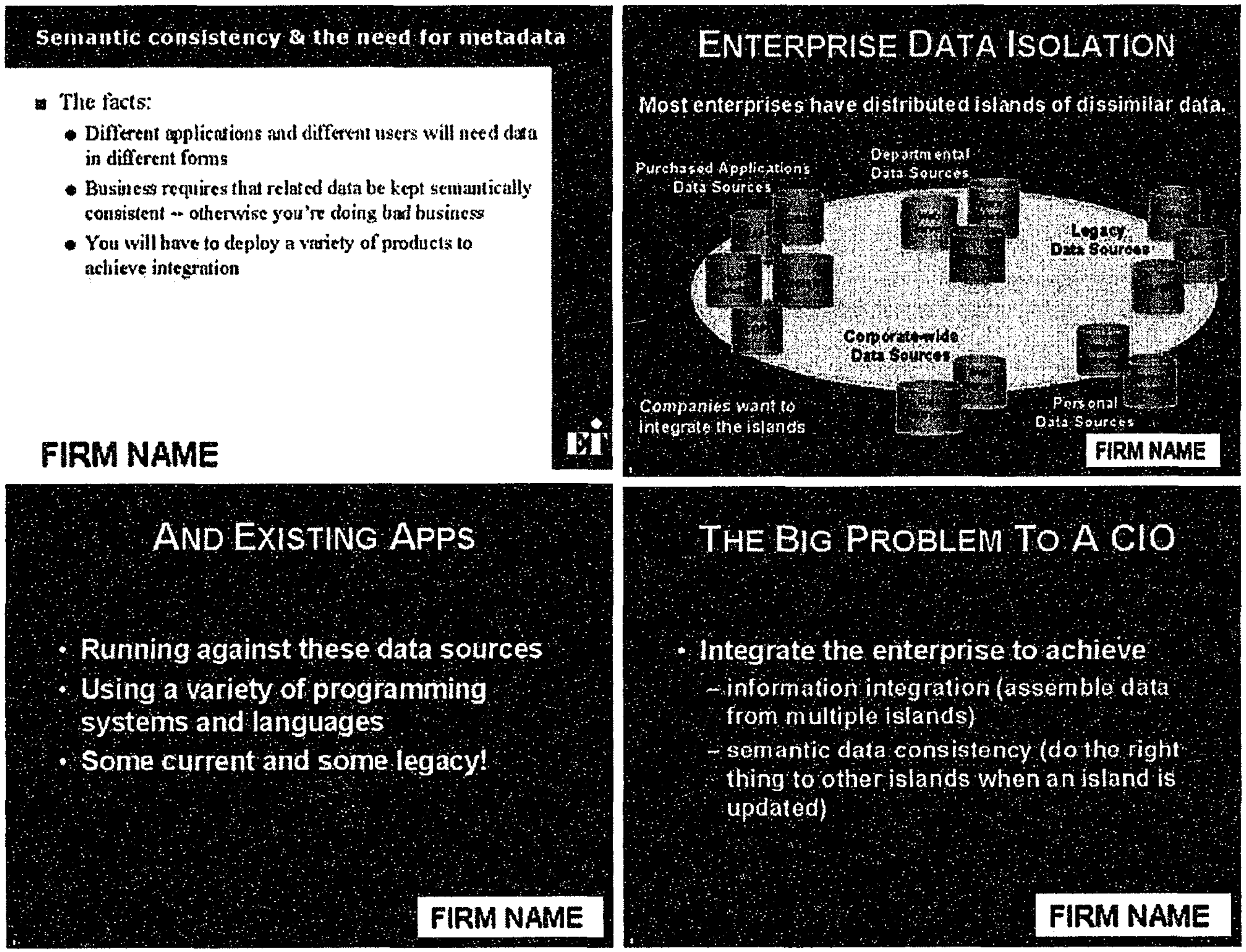

Figure 10: Firm's IS Issues

Source: Data provided by research participants

Data in Figure 11 depicts the key application integration issues identified by participants. In the absence of a common application integration platform, developers link applications by building individual communication links between applications. This leads to a combinatorial explosion in the number of links as new applications are added to the environment.

This explosion of links resulted in what one of the managers referred to as "islands of middleware" within the firm. FS1MD1 remarked that with this approach to application integration, "spaghetti is inevitable."

This approach also requires that application developers know multiple operating systems, programming languages and communications protocols. This leads to increased application development cycle times and cost.

To summarize, senior IS managers believed that this environmental complexity was the root cause of the high IS costs, increased application development life cycle and lower quality of service.

To seek relief, technology managers in organizations turned to middleware technology to address these issues.

These concerns dominate the data for the organizational context for middleware implementation.

To help solve such organizational issues, technology vendors attempted to create middleware standards that would meet the basic interoperability and data integrity needs of end-users. Since 1988, the technology industry has created three standards:

- DCE
- CORBA

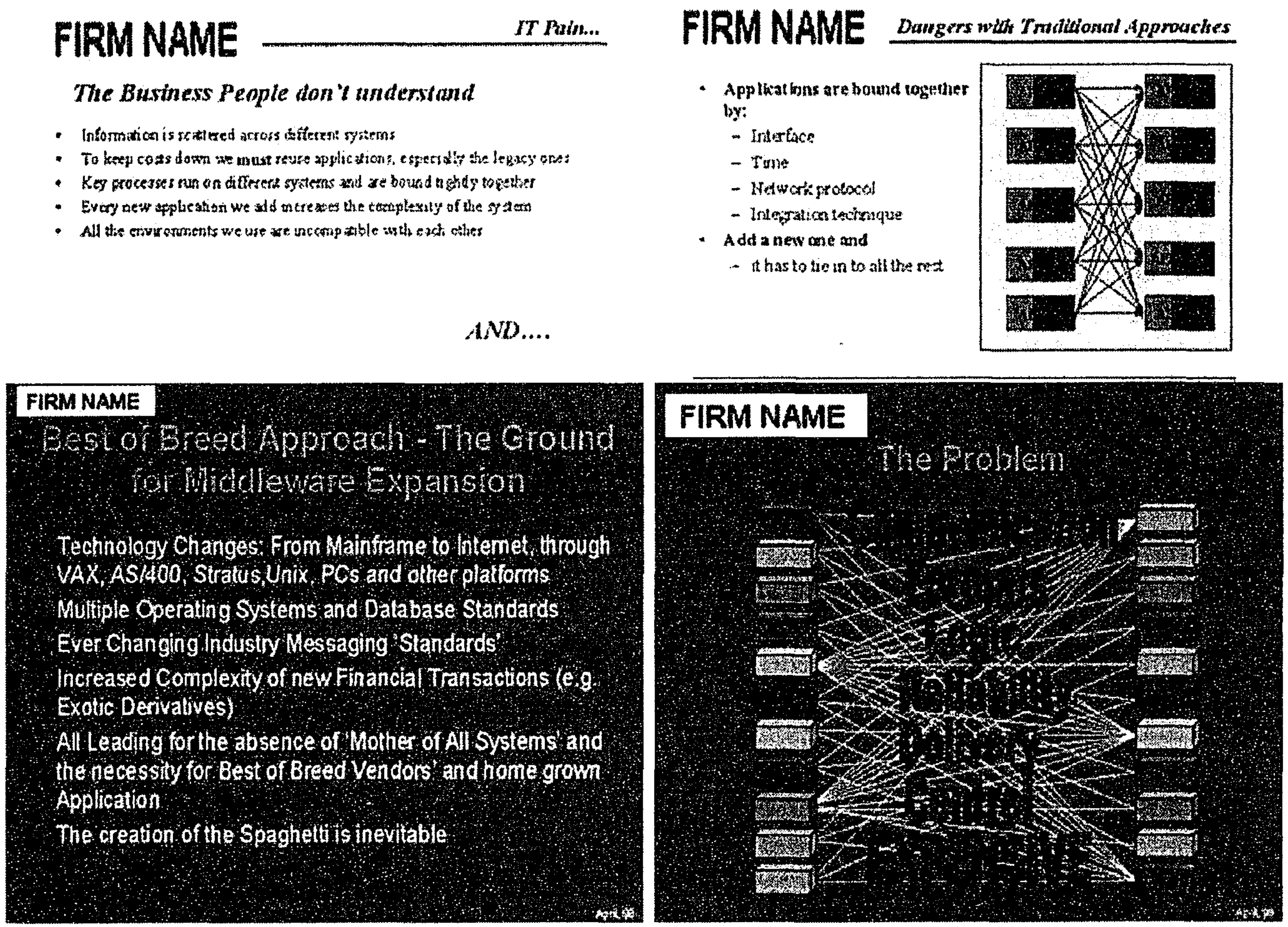

Figure 11: Application Integration Issues

Source: Data provided by research participants

- Web Services.

The following sections document the implementation of each of these middleware standards in organizations and the technology context that influenced their success or failure. The data is organized into the categories previously described in Tables 15 and 16, and correspond to the technology implementation process and technology context depicted in the middleware implementation framework given in Figure 7.

DCE

In the early 1990s, technology managers in several large firms made a strategic decision to adopt DCE technology to address their IS issues. This section describes the implementation of DCE and the technology context that influenced the failure of the implementation.

DCE Implementation Process

The data in this section corresponds to Theme II described previously and describes the implementation of DCE middleware in organizations. As depicted in the middleware implementation framework given in Figure 7, this process is coded as the core category called middleware implementation process and is segmented into the following four sequential phases:

- DCE technology strategy
- DCE implementation
- DCE in use
- Consequences.

These four phases are coded as sub-categories that are described in detail in Table 15. The following sections provide the data related to DCE implementation organized according to these sub-categories.

DCE Technology Strategy

Technology managers inside large organizations had a clear and broad vision of what needs the DCE middleware technology would fulfill and the benefits it would bring to the firm.

Technology managers, for example, expressed a strong need to simplify the distributed infrastructure. As FS1VP1 commented:

> There was a clear need to rationalize the [distributed] infrastructure and provide an integration platform that was independent of hardware and software [operating systems] layers.

As the data in Figure 12 shows, technology managers believed that DCE middleware would provide that platform and help integrate data and applications. They expected this infrastructure to eliminate the need for point-to-point application connections and "untangle the spaghetti." DCE was expected to provide standard mechanisms for communications and data interchange between applications. This would enable simple and cost effective application development as developers would have to learn only one application programming interface (API).

Figure 13 further elaborates the benefits these managers expected to reap from the centralized DCE middleware infrastructure. A centralized infrastructure would provide operational benefits like integrated security, location transparency and better application monitoring and management. They expected the total cost of ownership to decrease and

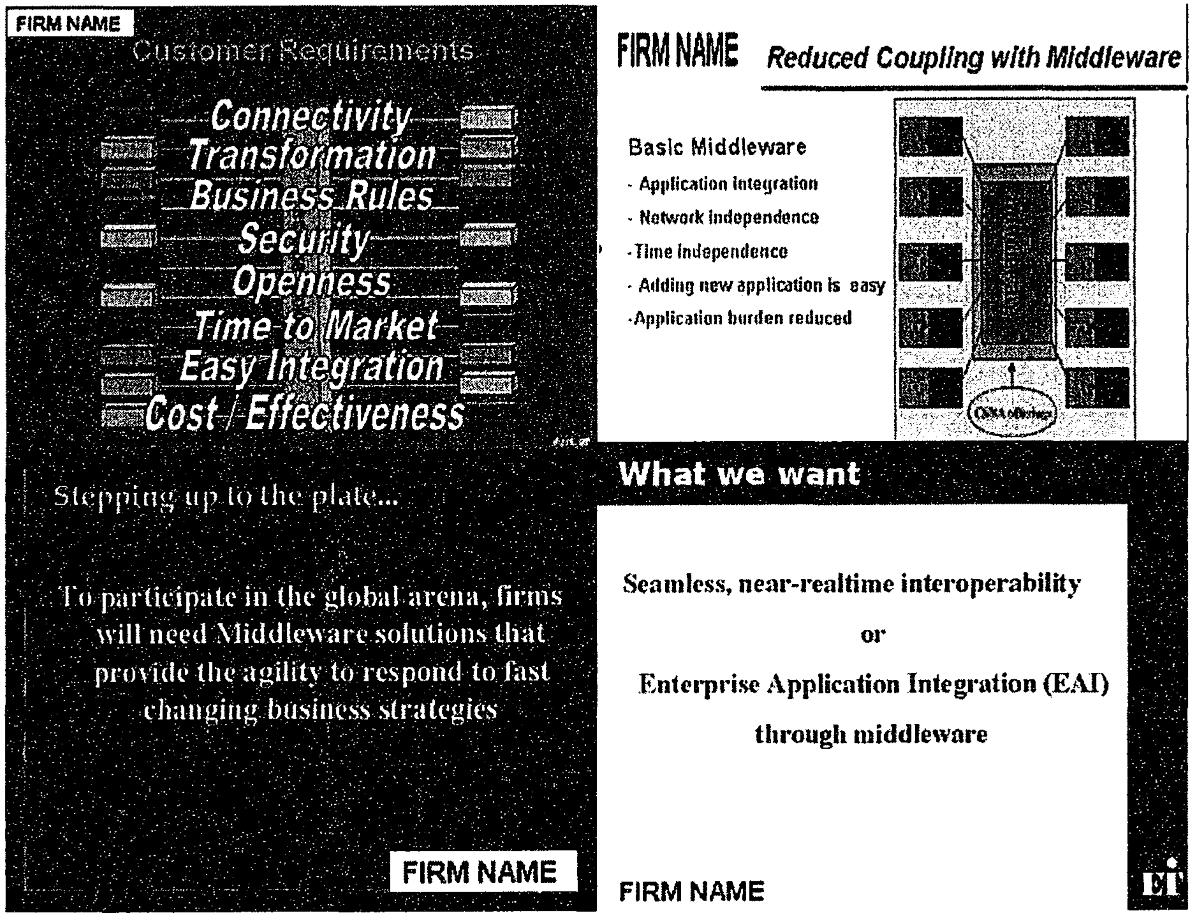

Figure 12: Infrastructure Rationalization by DCE

Source: Data provided by research participants

- Operational benefits
  - dramatically increased efficiency: lower unit costs, increased capacity, decreased cycle time
  - reduced risk
  - maximization of synergies across all business lines
- Business benefits
  - impact directly to bottom line
  - improved execution capability for business growth (i.e., faster time-to-market for new products and services)
  - additional capital for value-added investments

Figure 13: Benefits of DCE Middleware

Source: Data provided by research participants.

quality of service to increase as applications would leverage a common infrastructure. They believed this infrastructure would provide improved execution capabilities for business growth and maximize synergies across all lines of business.

Based on these assumptions and expectations, senior IT managers made a strategic decision to adopt DCE middleware technology. As a senior industry analyst said:

> The majority of the initial purchases of DCE were strategic enterprise-wide decisions based on the expectation by those at the highest levels of corporate IS organizations that DCE would be supported by all of the leading technology vendors across all popular technology platforms and therefore become a universal glue for heterogeneous distributed computing.

All technology managers in the three large global firms studied in depth pursued a well defined strategy to implement DCE in their firms. The next section documents the implementation of DCE in these firms.

DCE Implementation

The data indicates that the three firms studied deployed global DCE infrastructures sometime during the years 1995-1996.

The deployment of a global infrastructure in a large organization is a significant expense and risk. Following the decision to adopt DCE middleware, the technology managers in all the three firms studied adopted a well thought out implementation strategy. The data reveals that the managers in the firms studied took the following three kinds of actions to ensure the success of DCE technology in their organizations:

- restructured the engineering and support organization
- provided training and support to the application development teams

- tried to steer the industry direction to ensure continued development and support of the DCE standards.

Though these firms had been using global applications, none of them had deployed a global infrastructure before. This required extensive training of the engineering and support teams in the regional offices and reengineering of the tools and processes. FS1VP1 commented:

> We were the first ones to create the 'follow the sun model' for global support in the organization. The procedures and policies we developed at that time are still being used today for support the global applications that are being deployed now.

In some cases it required extensive reengineering of the support technologies as evidenced by the following comment from FS1VP1:

> The monitoring and problem tracking systems that were being used in UK were different from those in the US. But if you are supporting a global infrastructure, all the teams in the world need to be able to see the same information in real time. We reengineered these systems so the problem tickets opened in UK were automatically transferred to the US systems when the UK went to bed.

The senior managers carefully selected applications to pilot the technology and conducted extensive training classes for developers. They believed that these pilot projects would demonstrate to developers the advantages of the centralized infrastructure and that would propel the adoption of DCE by the entire application development community.

The three firms studied formed a financial industry coalition to address the industry-wide common issues with DCE technology. They used this forum to promote the DCE technology in other firms in the financial sector. They also used their collective purchasing power to encourage vendors to make enhancements to the technology and further promoted their requirements in the standards creation organizations. FS3MD1 commented:

> Once we had invested so heavily in the DCE infrastructure, our next concern was to protect our investment so that we could continue to invest with confidence. For that it was crucial that the vendors continue to evolve the technology, integrate other products with DCE and promote it as an industry standard...[this coalition] was the forum to make sure that vendors heard a unified voice from the financial community and that they continue to advance the technology we had invested so heavily in.

FS3MD1 continued:

> We wanted a broader adoption of the solutions we have chosen for our firms. It was a signification investment in an infrastructure and we wanted to make sure that we could get a decent ROI on it... Creating an industry-wide technology mindshare was crucial for this.

The commitment of the managers to drive the DCE technology into the broader market place is evident from the fact that two of the participants from end-user organizations were elected as chair of the DCE standards committee.

In retrospect, the managers believed that they had taken all the "right steps" to implement the technology. When FS2MD2 was asked "what would he have done differently?" he commented:

> "Looking back, I don't think we could have done things any differently. We did all the right things – conducted training, had pilots, provide the application development teams with experts who could help them out, worked with the vendors to make sure that they did the right things."

The next section documents the use of DCE technology at these firms and the factors the impacted its failure in firms.

DCE in Use

By mid 1997, it become apparent that an expectation for the majority of application development teams to utilize the DCE infrastructure was not going to be met.

A few pilots were successful, but the critical mass necessary to justify the expense of a global infrastructure never materialized. Table 17 shows the number of pilots and the actual production applications that were utilizing the DCE infrastructure at the three firms by the end of 1997.

Table 17: DCE Deployment in Firms

Firm	No. of Pilots	Production Applications in 1997
FS1	7	1
FS2	10	3
FS3	5	0

The participants in this study identified the following reasons for the failure of the adoption of the DCE middleware infrastructure:

- Incompatibility with the firm's business environment
- Incompatibility with the firm's technology environment
- Technology complexity
- Developer acceptance.

These are discussed more fully in the following sections.

Incompatibility with the firm's business environment

DCE middleware could not be used to integrate inter-organizational business processes. As IA2SA1 noted:

> The world of business is, by its nature, a very heterogeneous place. Technologies like DCE and CORBA were built with the assumption that applications at each end of the connection were built using the same technology. That can never scale to a viable business model in the real world.

DCE required that all applications use the DCE programming model and infrastructure to be able to communicate and interoperate. This prevented its use in the applications that supported inter-enterprise business operations. As FS3MD1 commented:

> In the DCE world, you were either DCE or nothing! I mean, you could not communicate with anybody unless they used DCE too. That may have been ok within your firm, but you can't dictate what the other party uses. Same with CORBA.

Most firms realized that it was not feasible to use DCE technology for inter-enterprise operations. For example, application development teams had a negative reaction to the high cost and effort required to use DCE. FS1VP1 commented on the difficulty of making the different groups within the firm use DCE to interoperate:

> We couldn't even make different groups in our firm use DCE to interoperate! The time and effort required to rewrite all the applications to use DCE was not worth it. These applications were working fine and to change them to just conform to a cleaner architecture was something we had a tough time selling [to the application development teams].

Most of the application groups within firms balked at the requirement that they had to do a major rewrite of their existing code to be able to use DCE to integrate with other applications. They argued that they were already interoperating with the applications they needed to communicate with and that the corporate requirement to use DCE to do something that was already being done did not justify the expense and the effort.

Incompatibility with key existing technologies at the firm

DCE did not integrate with some of the key technologies already in use at the firms. FS1VP1 commented:

> We use Oracle [database] all over the place. But [Oracle] did not adopt DCE [standards]. Most applications couldn't use DCE without Oracle. We tried helping [the application developers] but it was just too complex…. Firewalls were another story! [Since] DCE end points were dynamic, the ports changed each time the server's rebooted and then it wouldn't go through the firewalls! Of course, you could code the end points to be static [in DCE], only if the [application developers] could figure out how!

Databases are a key component of most corporate applications. The DCE standard was not adopted by the key database vendors including Sybase and Oracle. These vendors made little attempt to integrate their products with DCE and application developers were left on their own to figure out the complex task of using these products in their applications.

Firewalls, another key component of corporate infrastructure, had similar issues. DCE communications protocol was not compatible with the firewalls and hindered inter-enterprise communications.

In April 1997, a leading industry analyst predicted in a written report (Rymer, 1997):

> The Distributed Computing Environment (DCE) will fall out of favor during the next 12 months as a comprehensive "strategic architecture" for enterprise integration. …Organizations that try to use DCE as their strategic architecture will be forced to build most of their own applications and forego packaged software. For most organizations, this is an unsustainable commitment to custom development. Major ISVs (Independent System Vendors) have not supported DCE during the past and will find little reason to start supporting it during the next 24 months.

This lack of integration with the key technologies being used at the firm prevented the wide-spread use of DCE middleware in corporate applications.

Technology Complexity

DCE was a complex technology that was difficult to understand and use. One of the developers who worked for FS2MD1 gave an example of the complexity of the programming API:

> "As an example of the complexity of DCE, look at this function prototype from the DCE Application Development Reference:
>
> ```
> void sec_rgy_attr_sch_aclmgr_strings(
>         sec_rgy_handle_t context,
>         sec_attr_component_name_t schema_name,
>         uuid_t *acl_mgr_type,
>         unsigned32 size_avail,
>         uid_t *acl_mgr_type_chain,
>         sec_acl_printstring_t *acl_mgr_info,
>         boolean32 *tokenize,
>         unsigned32 *total_num_printstrings,
>         unsigned32 *size_used,
>         sec_acl_printstring_t permstrings[],
>         error_status_t *status);
> ```
>
> This [function] call has 11 arguments, 7 different types, none simple, some pointers, and one array. It will take you over a day to figure out what each of them really is! One look at the manuals will tell you that most of the calls are like this one."

The same developer pointed out that one had to read over a thousand pages of manuals just to understand the sample code provided with software.

The complexity prevented easy trials of the product and required a very steep and expensive learning curve. FS1MD1 noted:

> The biggest problem with [DCE] was [that] to do the simplest thing there was an awful lot of baggage. So there was a very steep learning curve. [It] wasn't easy to get into. One of the nicest things about http and web servers, [both components of Web Services,] is that you can dip your foot into water relatively easy with something like a CGI program.

The DCE system administration interface was equally complex. One of the system administrators who worked for FS1VP1 remarked:

> DCE has a few hundred commands for each component. Most of the time we couldn't figure out how to do some basic jobs like directory replication! It was very frustrating.

The operations teams were faced with the difficult task of learning hundreds of complex commands that were used to manage the infrastructure.

From the view point of developers and system administrators, the complexity of the product prevented its widespread use in the corporations.

Developer Acceptance

Other than the complexity of the product, the data indicated that there were two other reasons why the application developers resisted the technology.

First, DCE introduced an interface-based coding paradigm that developers were not used to. FS1MD2 commented:

> I don't think people had in their DNA to actually develop some sort of contract programming style where you describe your interfaces. Even if you didn't use DCE, but [if] you had a strong process of describing what was expected from a certain interface, results of the inputs and the outputs, [it] would have helped. I also think we were pushing hard on the infrastructure that was promised all things to all people but the people hadn't gone through the cultural change of understanding of what it means to really develop contract based programming styles where the interfaces are well understood and the expectations of the interfaces are well understood.

Even though developers had been writing client-server applications for a few years, they were not used to the discipline of describing their programs in terms of well defined inputs, outputs and exceptions. Senior IT managers and the industry saw this change as a positive one that would have enhanced the quality of the infrastructure, but developers were not ready for it.

Second, the DCE technology never really gained acceptance in the industry and that limited the career prospects for developers with DCE skills. As FS1VP1 commented:

> I can't blame the developers for lack of interest in the [DCE] technology. If the technology is not popular in the market, what's the point of spending months acquiring the skills which you can't use in your next job?

Developers had little incentive to learn a technology that would not improve their career prospects.

The incompatibility of the DCE middleware with the firm's business and technology environment coupled with its complexity and the limited adoption by developers had several consequences. These are described more fully in the following section.

DCE Implementation Consequences

The implementation of the DCE middleware and its limited use by users had several consequences, both organizational and technological.

The data from industry analysts and consulting firms revealed that most firms who attempted to deploy DCE had decommissioned the technology by 1999.

FSF3 decommissioned the DCE infrastructure at the end of 1997. FSF2 decommissioned the global infrastructure but retained just enough servers to serve the only application that was using it in production – this reduced DCE to be a part of the application rather than an infrastructure. This application was retired in 1999 and was replaced by a Java application. Since FSF1 had deployed a corporate-wide messaging system on DCE infrastructure and it was being used by several key applications, they were compelled to support the global infrastructure. This system was migrated to IBM's MQ technology in May, 2003 and the DCE infrastructure was decommissioned.

Organizationally, these experiences shaped managers' strategy for future middleware technologies and their approach to industry consortiums and vendors.

The managers realized that the approach to using a centralized middleware infrastructure was fundamentally flawed because it could never scale to a viable business model. As FS1MD1 commented:

> The simple fact is that oneness is bad. Centralized management is bad with Transaction Monitors and with workflow and Middleware because you cannot scale it to a viable business model.... you may still want a homogenous middleware but I believe that it is a myth. It is never going to happen. We should be looking at how to manage heterogeneity.

This realization shaped their attitude toward adoption of future middleware technologies. For example, as the data in the following sections shows, these organizations did not deploy CORBA as an infrastructure and it was used primarily by applications as a component.

The managers concluded that what is really needed is a technology that can interconnect "islands of middleware" because in a business situation, you can never really dictate what technology the partners and customers should be using to do business with your organization. FS1MD1 suggested that to be successful, middleware standards would have to be defined at the protocol level. He noted:

> Look at all the exchanges. There is a huge level of diversity of technology and huge amount of interconnectivity and huge number of companies. How do they deal with it? They give you a socket level protocol and it always works. And these are the hugely successful pieces of middleware. Having something that works at a protocol level is a huge benefit.

Their approach towards vendors and industry consortiums changed too. FS1VP1, who had actively participated in the OSF meetings noted:

> It was really an interesting experience [working with the industry consortiums]. One would assume that if IBM, HP, and all the other heavy-weights agree to a standard, well then it's a standard! Couldn't be more wrong.... Doesn't meant that they will make products that will work, doesn't mean they will continue to invest in the technology, doesn't mean their stuff would interoperate.

These companies realized that the adoption of a standard by an industry consortium does not translate into viable products that vendors will support in the long run. It also does not mean that the rest of the technology industry will follow those standards.

Though the senior managers in all the three firms that participated in this study were actively involved with OSF and OMG, the consortiums that were leading the DCE and CORBA efforts, none of them are currently actively involved in the Web Services standardization consortiums. As FS1VP1 commented:

> Why waste your time [in industry consortiums].... Let them figure it out. We'll use the stuff [middleware] if and when it is ready.

These managers had a "wait and see" attitude towards the industry consortiums that had repeatedly failed to reach consensus on technology standards.

In 1998, the DCE standards committee announced that they were formally ceasing any future work on the DCE technology and made the DCE code freely available to the industry. By 1999 most DCE vendors had stopped development work on DCE.

This completes the description of the data for the implementation process for DCE.

The next section describes the technological factors that impacted the implementation of DCE technology in organizations.

DCE Technology Context

The data in this section corresponds to Theme III described previously and describes the technological factors, external to the organization, which influenced the implementation of DCE middleware in organizations. As depicted in the middleware implementation framework in Figure 7, this theme is coded as the core category called technology context with the following sub-categories:

- DCE Technology standardization process
- Acceptance of DCE standards by major vendors
- Availability of complementary products.

These sub-categories are described in detail in Table 16 and the following sections provide the data related to DCE technology context organized in these sub-categories.

DCE Technology Standardization Process

The data indicates that the process by which the DCE standard was created was not optimal. Some of the problems that users encountered, which were detailed in the previous section, can be traced back to the standardization process.

DCE technology was produced by Open Software Foundation (OSF), a vendor consortium. OSF identified the technology components needed for a distributed environment, selected the technologies from participating vendors, integrated them into a product and defined that as the standard (Open Software, 1995).

The participants in this research indicated that this process ignored some of the key factors that have made other technology standards a success.

The members who participated in the OSF standardization committees were typically vendor engineers and technologists who did not have any background and experience with user organizations. As FS3MD1 commented:

> Most of the [participants in OSF meetings] were [vendor] engineers. They are pure technologists. They had no clue of the business problems we were trying to solve. When I joined OSF, [vendor engineers were] amazed that we actually used this stuff for mission critical [business] applications! And when I described the issues we were facing, they were stunned!

This lack of understanding of the user environment and problems led to the selection of technology components that were not suitable for use in a complex global organization. FS3MD1 commented on the selection of the directory services component of DCE:

> The [OSF participants] didn't have enough experience in the field with the technology to see which technology was good and which one was bad. So they chose badly. If they had experience in the field with end user systems, they would have realized that HP's Domain Location Broker was a dynamite

> product and CDS was a piece of junk. If you choose standards before actual products have evolved and [have been] tested in the field, then you end up choosing the wrong ones.

The technologies incorporated into the DCE standard were immature and had not evolved with the benefit of field experience. FS1MD1 commented that "DCE was a case of premature or inappropriate standardization." This occurred because the technology was declared a standard, and therefore it could not evolve as rapidly as other technologies that were not standardized and are therefore free to evolve.

The participants indicated that the OSF process was not open to the broad technology community and so individuals who had special expertise and know-how did not participate in the process. FS1VP1 commented further:

> Internet standards are a perfect example of how standards should be created! Everyone gets a chance to review it, try it out, point out errors, issues.... Both OSF and OMG did none of these.

The result of this closed process was a standard that did not meet the needs of user organizations.

Acceptance of DCE Standards by Major Vendors

The lack of support by key vendors impacted the adoption and implementation of DCE middleware. The following excerpt from an analyst report describes the situation (Rymer, 1997):

> At the March 1997 meeting of Open Group members, users and vendors debated a familiar topic. How can users persuade intransigent software vendors to support DCE? 'The vendors should be flogged,' quipped one member. Several vendors countered that they had little confidence that investments in DCE would generate returns in the market.

Users were uncomfortable using a technology that was not supported by the key industry vendors. Even the vendors who were part of OSF did not support DCE in all the other products they were offering to users.

FS1VP1 identified the reason for lack of commitment by major vendors. He noted:

> Technology companies did put their bet on different products and standards. IBM and HP put their bet on multiple products – they supported both DCE and CORBA and some others. Sun had its own brand of middleware. After all they were [at that time] the only ones with a successful middleware story with RPC, directory services and distributed file systems. Microsoft was brewing COM and other stuff….These [vendors] had no reason to converge to a single standard.

Both Microsoft and Sun Microsystems did not see any benefit in supporting the DCE standards. Though Sun Microsystems joined OSF in 1993, it refused to support DCE and continued to support its own technology. FS3MD1, who chaired the DCE committee, noted:

> The only goal the Sun rep [in OSF DCE committee] had was to derail the standards and agreements. Every one was suspicious of him. It was a big furor when I invited him to sit next to me in the inaugural session.

The industry analyst report quoted above (Rymer, 1997) further predicts the future of DCE:

> This impasse helps explain why DCE will have a limited future as comprehensive enterprise architecture. Undoubtedly, DCE is a robust architecture for distributed computing. But weak industry support for DCE makes it impractical to adopt. Therefore, DCE will fall out of favor as a comprehensive "strategic architecture" for enterprise integration during the next 24 months. Some users will continue to use DCE to support individual applications, particularly those involving transaction processing. However, the

> dream of using DCE as common systems services linking applications across enterprises is over.

This lack of support by major vendors also impacted DCE support by other vendors who provide complementary products. This is described more fully in the next session.

Availability of Complementary Products

The data indicates that lack of complementary products prevented the widespread adoption of DCE in the user organizations.

The following excerpt from an analyst report describes the situation (Rymer, 1997):

> So far, ISVs have sealed the fate of DCE. Most ISVs do not support DCE services today, and they will not support those services during the next 24 months either. DCE faces an up-hill battle for acceptance among ISVs.

Most ISVs, an industry termed used for smaller independent software vendors, did not support DCE and did not integrate their products with the technology. Typically, ISVs produce complementary products for technologies that have wide-spread adoption and use. They did not believe that investments in DCE technology would generate viable returns.

The same report continues to describe the problems:

> Organizations that try to use DCE as their strategic architecture will be forced to build most of their own applications and forego use of packaged software. For example, the Jet Propulsion Laboratory (JPL) has made DCE its standard architecture, but has been unable to find basic software that supports it, says Steven Jenkins, deputy manager and chief engineer of Enterprise Information Systems. Thus, JPL chose Oracle Financials even though the package doesn't support DCE. Jenkins also can neither find a group calendar package that

> supports DCE, nor persuade JPL's incumbent vendor to add even partial DCE support.

The participants from user organizations reported similar issues. The key database and firewall vendors did not support the DCE standards. Since integration with these products is crucial for enterprise operations in large organizations, it impacted the adoption of DCE in most large organizations.

FS1VP1 noted:

> You cannot disregard databases, firewalls and other packaged software [for middleware]. Most applications need these and since DCE did not integrate with these, it was not of much value.

Lack of complementary products limited the usefulness of DCE middleware in large organizations and was one of the significant causes for its limited adoption.

This completes the description of the data for DCE middleware. The following section provides a brief summary.

DCE Data Summary

> DCE did not fulfill its initial ambitions of a universal platform for distributed computing. The majority of the initial purchases of DCE were strategic enterprise wide decisions based on the expectation by those at the highest levels of corporate IS organizations that DCE would be supported by all of the leading technology vendors across all popular technology platforms and therefore become a universal glue for heterogeneous distributed computing. That promise did not materialize. - Yefim V. Natis, VP, Research Director, Gartner Research, 1999

The above quote from the research director of a prominent industry analyst firm aptly summarizes the fate of the DCE technology in large enterprises.

From the data it is evident that external technological forces, particularly lack of adoption by key industry vendors and a dearth of complementary products, limited the usefulness of DCE in large organizations.

DCE middleware did not meet the needs of user organizations. Its failure impacted the strategy the managers adopted for CORBA, the next generation of middleware.

The following sections describe the data for CORBA middleware.

CORBA

CORBA was produced by a vendor consortium called Open Management Group (OMG).

Even though OMG and OSF started the developments of CORBA and DCE around the same time, a stable and usable version of DCE was available to users a few years before CORBA. The experience of users with DCE, documented in the previous section, influenced their strategy towards the use of CORBA.

The following sections document the implementation experience of CORBA in organizations and the technological factors that influenced its failure, as reported by participants in this study.

CORBA Implementation Process

The data in this section corresponds to Theme II described previously and describes the implementation of CORBA middleware in organizations. As depicted in the middleware implementation framework in Figure 7, this process is coded as the core category called middleware implementation process and is segmented into the following four sequential phases:

- CORBA technology strategy
- CORBA implementation
- CORBA in use
- Consequences.

These four phases are coded as sub-categories that are described in detail in Table 15 and the following sections provide the data related to CORBA implementation organized into these sub-categories.

CORBA Technology Strategy

Unlike DCE, there was no strategic enterprise-wide decision to deploy CORBA in a systematic way to address the middleware IS issues. FS1VP1 noted:

> Thank God we never deployed CORBA as an infrastructure here! After DCE, we had no appetite to go for [middleware infrastructure] … [application development teams] used it on their own as they saw fit.

The failure of DCE as an enterprise middleware platform had made users aware of the limitations of a centralized middleware infrastructure. As a result, even though CORBA had all the necessary services to serve as a common infrastructure, it was deployed as a component of applications.

Individual application development teams made their own decision to use CORBA based on their needs and requirements. FS1VP1 noted further:

> Application development teams used CORBA primarily as an Object Oriented tool. It was good for defining interfaces and brought in some programming

> discipline. Several applications here used it, but there was no coordinated effort.

The decision to use CORBA was driven primarily by the need to simplify client server application development when using Object Oriented programming language like C++. Application development teams were keenly interested in the benefits of using an object oriented technology and any benefits of using CORBA were incidental.

There were no expectations at the higher levels of management that CORBA would simplify the infrastructure or solve enterprise-wide data and application integration issues.

The next section documents the implementation of CORBA in organizations.

CORBA Implementation

Since there was no corporate level strategy for CORBA in large firms, its implementation was left to individual teams. There were limited deployments of CORBA in firms and over time it became a small component of large application projects. As depicted in Figure 14, this limited implementation of CORBA was a concern in OMG's technical committee meeting in June 2002.

Unlike the case with DCE implementation, senior technology managers did not have firm-wide plans for CORBA implementation.

Since CORBA was not deployed as an infrastructure, there was no need to modify the processes and systems for supporting its deployment. CORBA was supported as part of applications. As one of the infrastructure managers commented:

> We were still pushing DCE as the middleware in those days. When some of the [application development teams] started using CORBA, we treated it as just

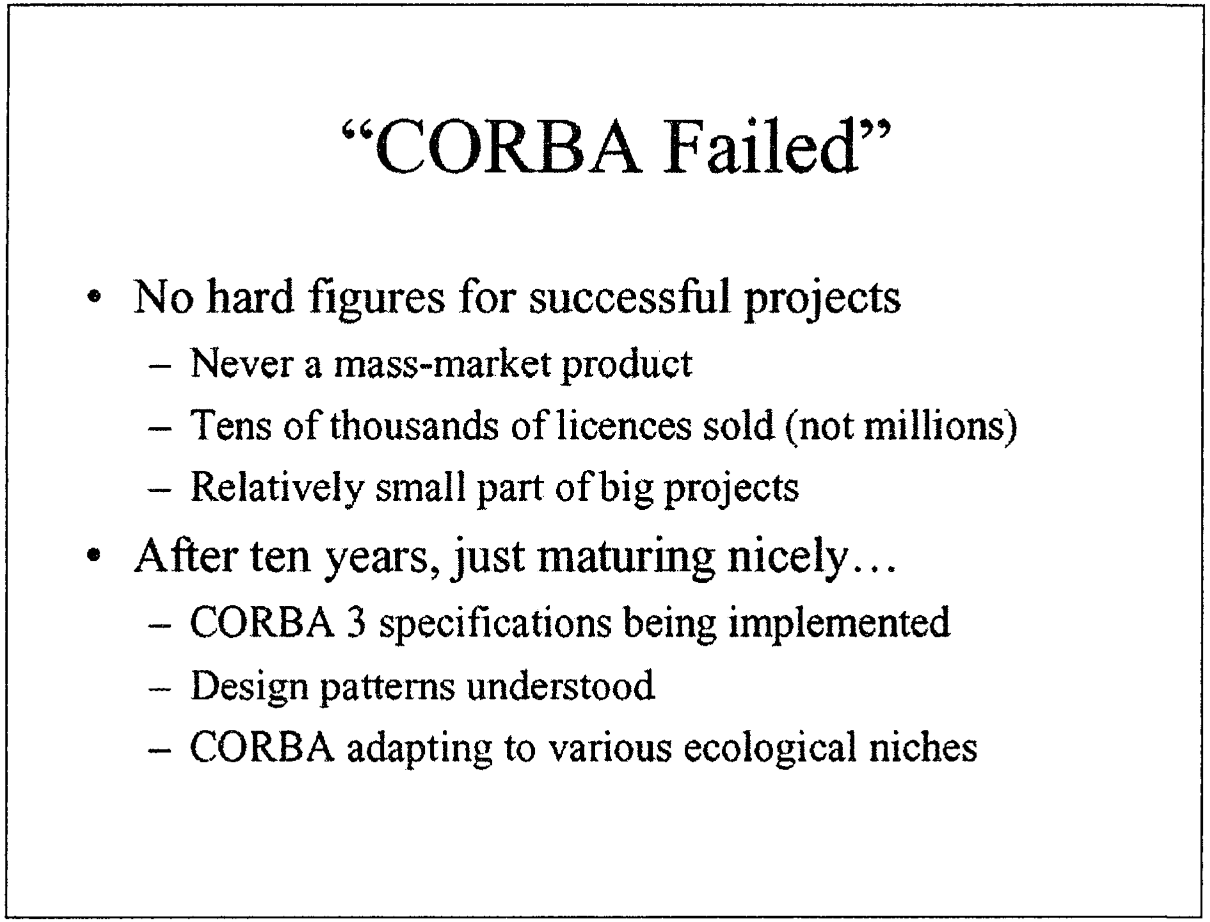

Figure 14: CORBA Implementation in Organizations

Source: OMG Technical Committee Discussion (Welsh 2002)

> another application. We did try to work with OMG and OSF to merge the specifications but it did not work out.

Since CORBA specifications were being developed almost in parallel with DCE, technology managers at the three user firms that participated in this study made an attempt to integrate two standards, but OMG rejected the DCE communications and security protocols.

Unlike with DCE, study participants could not give exact figures about the extent of CORBA use in their firms, but they all believed that the number of applications in use at their firm was less than a hundred. This was in alignment with estimates provided by industry analysts and consulting firm participants.

The next section documents the reasons for the limited use of CORBA in organizations.

CORBA in Use

DCE and CORBA had similar architectures and suffered from some of the same limitations that restricted their use in organizations. The data in Figure 15, taken from an OMG Technical Committee meeting, identifies some of the issues.

The participants identified the following reasons for the limited adoption of CORBA products in end user organizations:

- Incompatibility with firm's business environment
- Incompatibility with firm's technology environment
- Product complexity
- Developer acceptance.

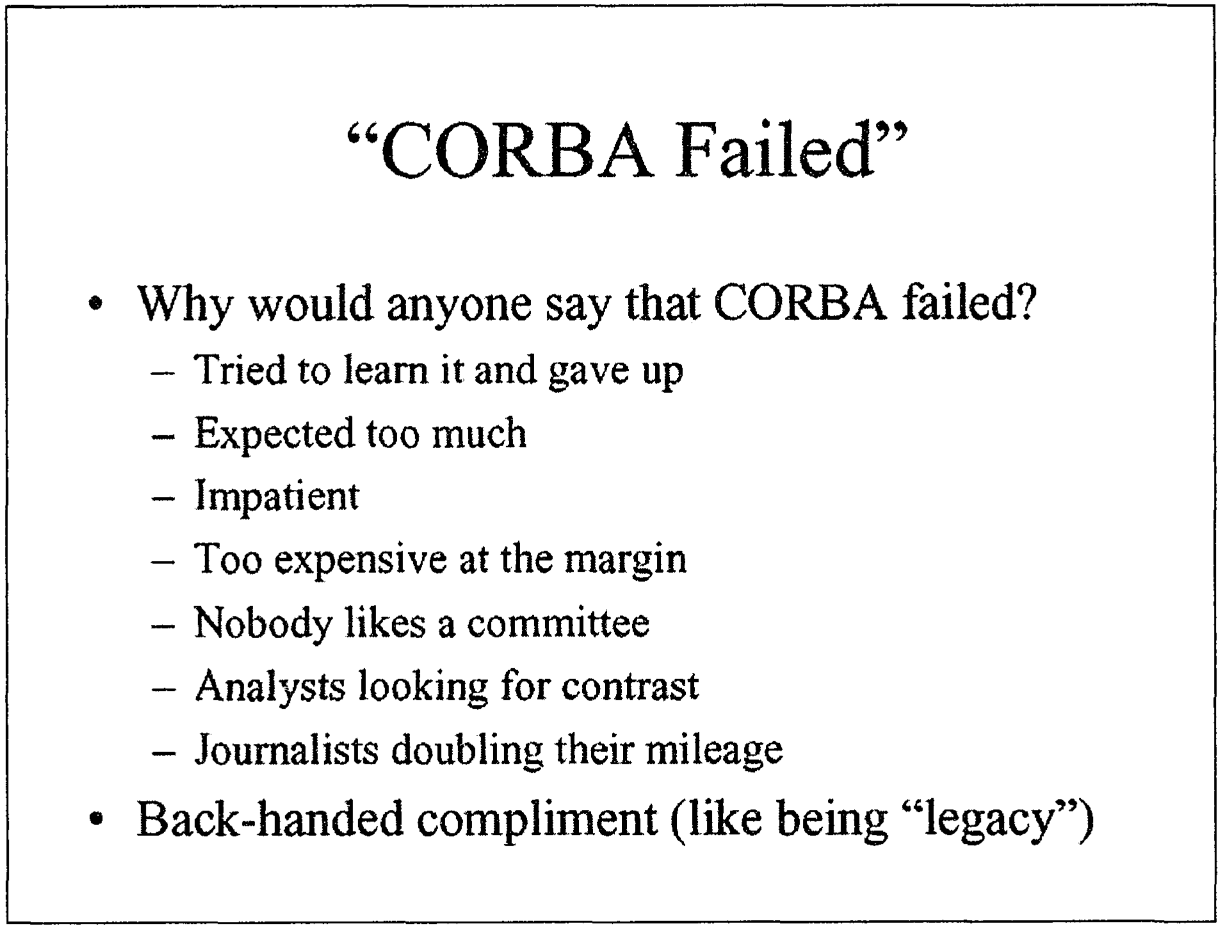

Figure 15: Reasons for CORBA Failure

Source: OMG Technical Committee Discussion (Welsh, 2002)

These are discussed more fully in the following sections.

Incompatibility with Firm's Business Environment

The following quotes, repeated from the data for DCE implementation, identify the issues with using CORBA in a business environment:

> The world of business is by its nature a very heterogeneous place. Technologies like DCE and CORBA were built with the assumption that applications at each end of the connection were build using the same technology. That can never scale to a viable business model in the real world.
>
> – IA2SA1

> In the DCE world, you were either DCE or nothing! I mean, you could not communicate with anybody unless they used DCE too. That may have been ok within your firm, but you can't dictate what the other party uses. Same with CORBA.
>
> - FS3MD1

CORBA, like DCE, required that all applications use the CORBA programming model to be able to communicate and interoperate. This prevented the use of CORBA in inter-enterprise applications and with legacy applications at the firm unless all applications converted to the CORBA platform. The following quote from an industry analyst report (Gilpin et al., 2001) reflects this limitation:

> Previous means of communicating between applications over the Internet were not sufficiently "Internet friendly," requiring that special provisions be made at the firewall and requiring a larger "footprint" of added technology at each end to effect the connection. Failed candidates included CORBA IIOP, Microsoft DCOM or various message-oriented middleware (MOM) products such as IBM MQSeries.

CORBA's inability to provide interoperability with applications and the high cost of migrating legacy applications prevented its wide-spread adoption by application development teams. The data in Figure 15 indicates this issue too.

Incompatibility with Firm's Technology Environment

CORBA is an object-oriented technology and did not integrate with the key database technologies in use – Sybase, Oracle and Informix. Database vendors made an attempt to create object-oriented databases but the complexity of the technology prevented the development of usable products. FS1VP2 commented:

> Some of the vendors tried creating OO databases – but it was just too complex. For a while, [the industry] tried creating the specifications for this but it never went anywhere. Eventually, everyone gave up.

The deployment of CORBA, like DCE, also required special provisions at the firewalls (Gilpin et al., 2001) and this also hindered its use for inter-enterprise applications.

Microsoft had a competing product called Component Object Model (COM) and it did not integrate CORBA with their desktop technology. Since Microsoft dominates the desktop platform, CORBA was not integrated with key Microsoft products which were, and still are, very widely used at firms.

Technology Complexity

CORBA, just like DCE, was a complex product to use. According to FS1VP1, developers complained about the "kludgy C language interface and about how difficult it was to understand how CORBA worked under the hood."

Object-oriented programming, particularly with languages like C++, is a difficult task and the complexity was compounded by the CORBA programming model. FS1VP1 added:

> It was not easy for most developers to understand interface-based programming [with CORBA]. It was a new concept for them. If you add to it the complexity of coding in C++, the whole thing is quite difficult. Only our star developers could handle this.

From the above quote, it was concluded that the programming interface was complex and counter-intuitive and this caused a steep learning curve. It was not easy to try out the product and build prototypes quickly.

The complexity of the product contributed to its lack of widespread use.

Developer Acceptance

The complexity of the product, coupled with the popularity of Java, prevented wide adoption of the CORBA technology by developers. The data in Figure 16 shows the effect of Java on CORBA. FS2MD1 noted:

> Java took the developer community by storm. They just loved it. Writing Object Oriented code with Java is so easy. No more issues of inheritance [as in C++ language], no dangling pointers once you delete an object....Suddenly, there was no need to use CORBA. Java had everything you needed.

The rapid evolution of Java from a programming language to an enterprise wide integration platform obscured CORBA. Developers lost interest in CORBA because Java was a very attractive development platform that integrated very well with the Internet and Web-based applications.

CORBA, like DCE, had a steep learning curve and was not freely available to developers. Java, on the other hand, could be freely downloaded over the Internet and developers could play with it. FS1MD2 noted that "CORBA never had a chance against Java." Java's growing demand within the user industry caused developers to move away from CORBA.

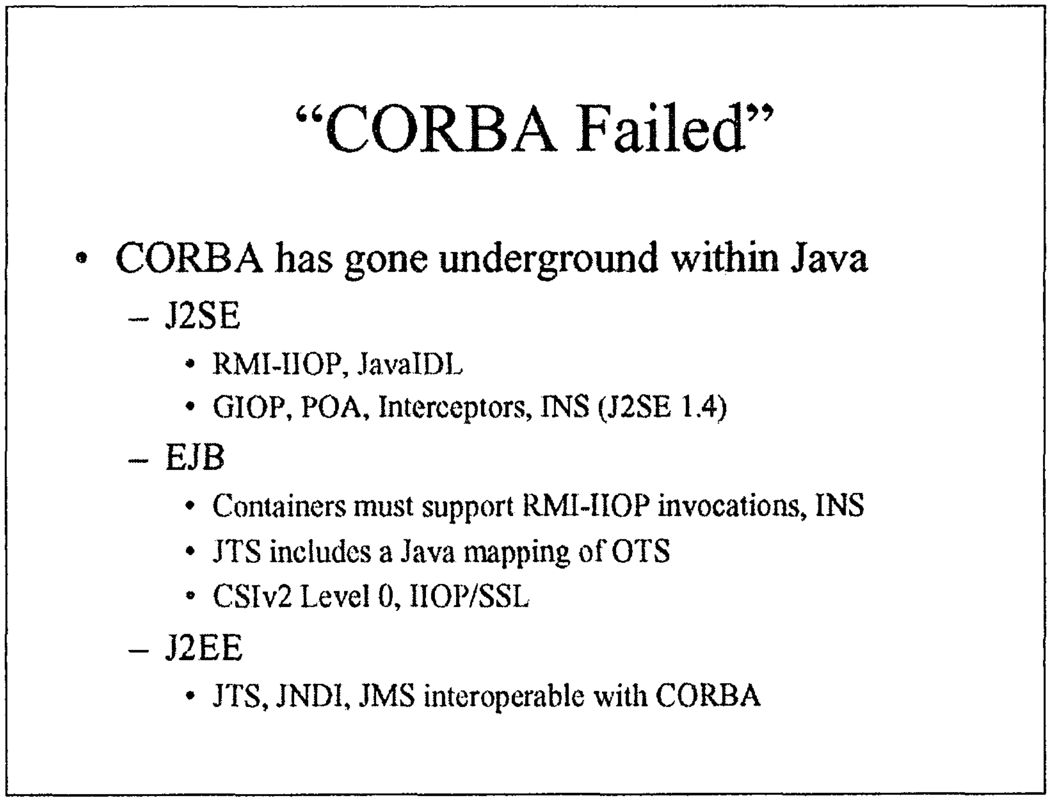

Figure 16: Java Overshadowed CORBA

Source: OMG Technical Committee Discussion (Welsh, 2002)

These factors led to limited use of CORBA in the industry. The consequences of this limited use of CORBA are discussed in the next section.

Consequences of CORBA implementation

CORBA is still in use at firms where it was deployed in the late 1990s, but none of the firms who participated in this research have developed any new application using CORBA since 2000 and have no plans to doing so. The interviews with analysts confirmed that this is the trend across the industry.

The limited adoption of CORBA had little or no organizational impact because the scope of its deployment was limited. FS1VP1 noted:

> There is no real impact here. We didn't have any infrastructure with CORBA, so there is nothing to undo when we phase it out. The applications that are using it, will eventually replace it with something else, I guess Java or any Web [Services] based product.

The managers who participated in this study indicated that most of the current applications using CORBA would phase out in the next technology refresh cycle. Most of the technology vendors who support CORBA are also encouraging their users to move off CORBA.

Though OMG continues to support CORBA, its focus has now shifted. The following excerpt from an industry analyst report (Allen, 2002) shows this change in focus:

> In 1999, the 800 member companies that make up the OMG began to consider moving beyond CORBA and contributing to a better solution of the integration problem. It was clear that CORBA would never be able to replace all of the other middleware standards companies were using.... Faced with this reality, the OMG began to consider a new approach to architecting distributed

> systems. In the late 1990s, the OMG had sponsored the development of an integrated software modeling approach, the Unified Modeling Language (UML)...After considering some pioneering work by member organizations, the OMG decided to move towards adopting a new architecture, which is named the Model Driven Architecture (MDA).

OMG decided to abandon the effort to create middleware standards and shifted its focus to creating methodologies for integration. The report continues:

> In effect, the OMG decided to raise the level of abstraction and focus on how the system should be integrated [rather than create a standard to do it]. This is not to suggest that OMG is abandoning CORBA. Rather, it will continue to support CORBA and its OMA (Object Management Architecture), while simultaneously developing a second, more comprehensive, UML based architecture.

All the major vendors who are part of the OMG consortium are now involved in developing the Web Services standards. These include IBM, Sun Microsystems, Hewlett Packard and Microsoft, among others.

This completes the details of the data related to the implementation process for CORBA middleware. The next section describes the technological factors that impacted the implementation of CORBA middleware in organizations.

CORBA Technology Context

The data in this section corresponds to Theme III described previously and describes the technological factors, external to the organization, which influenced the implementation of CORBA middleware in organizations. As depicted in the middleware implementation framework in Figure 7, this theme is coded as the core category called technology context with the following sub-categories:

- CORBA technology standardization process

- Acceptance of CORBA standards by major vendors
- Availability of complementary products.

These sub-categories are described in detail in Table 16. The following sections provide the data related to the CORBA technology context organized into these sub-categories.

CORBA Technology Standardization Process

Study data indicates that the process by which the CORBA standard evolved was not optimal. Some of the problems that users encountered, detailed in the previous sections, can be traced back to the standardization process.

Participants indicated that the CORBA standards were created without any experience in the field. FS1VP1 said:

> They (OMG) created a paper standard first and then vendors tried to create products to match those standards. No one knew whether the standards would work or not! They depended on the feedback from us to improve the product. But who has the time to do experiments on their behalf!

CORBA is a "paper standard" that was developed by members of OMG through debate and discussion. There was no experience in the field with real products that led to the creation of the initial standards. Each generation of products that were developed according to the specifications provided the feedback and experience necessary for the development of the next generation of specifications. FS1VP1 noted that it took "[OMG] forever to get something useful." The data in Figure 17 captures this in a cartoon that was part of the OMG Technical Committed discussion in June, 2002. This feedback loop, also evident in the data in Figure 18, delayed the availability of viable CORBA products.

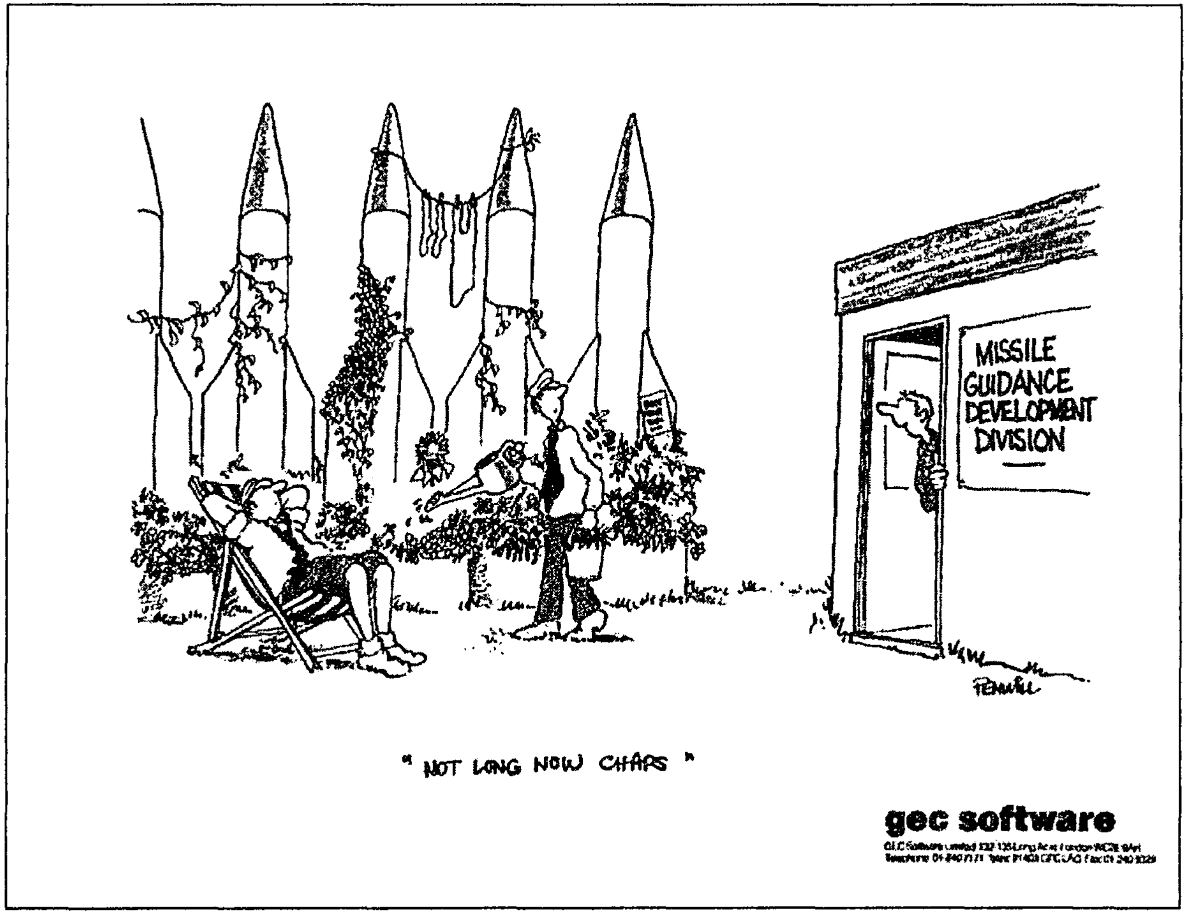

Figure 17: CORBA Standard Delays

Source: OMG Technical Committee Discussion (Welsh, 2002)

"CORBA Failed"

- OMG Process
  - As great an achievement as any specification
  - Arguably the best blend of standards and enterprise
- Multiple Feedback Loops
  - Application fails? Get training, mentor
  - Product fails? Call vendor
  - Specification inadequate? Revise at OMG
  - Process weak? Modify OMG process
- "Use standards as a ratchet" (Deming)

Figure 18: CORBA Standardization Process

Source: OMG Technical Committee Discussion (Welsh, 2002)

IA2SA1 commented that this "learning by doing" model and using feedback to improve the next generation of technology was a good way of gaining experience, but delayed the availability of technology to such an extent that it was irrelevant by the time it matured and was overshadowed by newer technologies.

Just like the DCE standardization process, the CORBA standardization process not open to the broad technology and end-user communities. Data in Figure 19 alludes to the limitations of this closed process. Without the benefit of a scientific review process, the standards that were created were limited in functionality and did not meet the needs of the users.

The delays in the creation of viable standards impacted the availability of products that could meet the need of users. This, coupled with the lack of support from key industry vendors, impacted the implementation of CORBA in organizations.

This is discussed in detail in the following sections.

Acceptance of CORBA Standard by Major Vendors

The lack of support of CORBA standards by Microsoft impacted its adoption and implementation. The following excerpt from an industry analyst report shows Microsoft's position (Moore, 1996):

> It is now apparent that Microsoft will not support the Object Management Group's (OMG's) Common Object Request Broker Architecture (CORBA) specifications for distributed object management. Instead, Microsoft will promote its own Enterprise OLE as an alternative to CORBA-compliant products... CORBA vendors must achieve critical mass among customers during 1996 if CORBA is to survive Microsoft's onslaught.

What Could We Do Better?

- Stop burying our failures
  - Possibly main reason for ongoing "software crisis"
- Openness is a prerequisite for progress
  - Scientific publication, peer review
  - Bridges, buildings, aircraft, etc.
  - Open source, open specifications…
  - …but secrecy about internal projects
- As at OMG, better to pool knowledge
  - Standardize what everyone is willing to share

Figure 19: CORBA Standardization Process Deficiencies

Source: OMG Technical Committee Discussion (Welsh, 2002)

All the key technology platform vendors except Microsoft joined the OMG. As discussed in the details for DCE, some vendors like HP and IBM supported both DCE and CORBA standards. Without a full commitment to a standard by the major vendors, the standards just fizzled out over time.

This disagreement among vendors, referred to as vendor politics in the extract below from an industry analyst report (Natis, 2001), prevented the development of CORBA standards:

> New vendor alliances redefine the competitive marketplace: The evolution of the application server market (as most markets) is significantly influenced by the vendor's politics. Java emerged as a phenomenon, in part, because it served as a unifying cause for vendors looking to compete against Microsoft; CORBA or the Distributed Computing Environment (DCE) never succeeded in the mainstream market, in part, because there was no established agreement between vendors to invest in those technologies.

This lack of support by Microsoft also impacted the availability of complementary products for CORBA. This is discussed in the next section.

Availability of Products that Meet the CORBA Standards

Study data indicates that, just like DCE, software vendors did not produce complementary products for CORBA. An excerpt from an industry analyst report (Natis, 2001) indicates the lack of productivity tools for CORBA:

> Despite its benefits, CORBA has never become a mainstream standard nor have popular productivity tools been created for it.

The Independent System Vendors had no incentive to produce tools and products for CORBA. As FS1VP1 noted:

> You have to realize that these vendors produce only for the mass market. It is not economically viable to make products for a small market like DCE or CORBA. We could never find management or application development tools that could help us with either DCE or CORBA. For DCE, we even funded some small vendors like UnixPros to write tools for us, but it did not go far – it was just not viable. For CORBA, we did not even try.

This lack of complementary products hindered the implementation and use of CORBA products in large organizations.

This completes the description of data for the technology context for CORBA middleware. The next section provides a brief summary of the data for CORBA implementation in organizations and the technology context that influenced it.

CORBA Data Summary

CORBA suffered from some of the same problems that limited the adoption and implementation of DCE, as evident from the following quote:

> "Like DCE before it, Common Object Request Broker Architecture (CORBA) has failed to become a universal platform for business applications...Leading CORBA vendor Iona Technologies have declared Orbix, its flagship CORBA ORB, an application integration solution [as opposed to an infrastructure solution]. OMG also takes a similar position."
> – Yefim V. Natis, VP, Research Director, Gartner Research, 2002

The CORBA standard failed to meet the needs of users. It was incompatible with their business model and with existing technologies in use at the firms.

This was a result of several factors. The standardization process was not optimal and delayed the creation of viable products. The lack of support by key vendors, particularly Microsoft, prevented its wide-spread use on popular desktop platforms. This also limited the support by ISVs who provide much needed complementary products like management and application development tools.

The failure of CORBA made technology managers more cautious in their approach to accepting middleware technology standards and influenced their strategy towards Web Services, the next generation of middleware standards.

The general consensus among participants in this research was that the DCE and CORBA technologies provided the learning experience behind the development of Web Services technology. All the participants also expressed the need to be cautious and not be overly optimistic about the possibility of Web Services being adopted by the general industry as a standard.

The data for the third major standard, Web Services, is discussed in the next section.

WEB SERVICES

Web Services are the third generation of standards in the evolution of enterprise middleware. Web Services standards are still evolving and the major vendors are working through various industry consortiums to converge on a set of viable standards. The following excerpt from an industry analyst report describes how Web Services differs from the previous attempts to standardize middleware technology (Lanowitz, 2003):

> At their very core, Web Services are a continuation and evolution of enterprise middleware. Plenty of technologies have served as the 'prequel,' including the distributed computing environment (DCE), Common Object Request Broker Architecture (CORBA) Internet Inter-ORB Protocol (IIOP), Java Remote Method Invocation (RMI) and components. What sets Web services apart from these other, mostly failed technologies is the agreement of vendors on standards. The acceptance and endorsement of standards, in theory, delivers proliferation in the creation of Web Services ...Proliferation of a technology

> drives the vendors to actually deliver on the hype; otherwise, the vendors' focus drifts to promoting the next great, hyped technology.

Users, who have seen the failure of previous industry attempts to standardize middleware technology, are cautiously optimistic and focusing on re-architecting their environment to ensure success with the new standards.

The following sections document study data that details the implementation of Web Services in organization and the technology context that may impact its success.

Web Services Implementation Process

The study data in this section corresponds to Theme II described previously and describes the implementation of Web Services middleware in organizations. As depicted in the middleware implementation framework in Figure 7, this process is coded as the core category called middleware implementation process and is segmented into the following four sequential phases:

- Web Services technology strategy
- Web Services implementation
- Web Services in use
- Consequences.

These four phases are coded as sub-categories that are described in detail in Table 15 and the following sections provide the study data related to Web Services implementation organized into these sub-categories.

Since the Web Services standards are still evolving, most organizations are in the process of developing their implementation strategy and there are limited pilot deployments of Web Services. So there is no data yet available on the last two sub-categories – Web Services in use and Consequences.

The following sections provide details from the study data.

Web Services Technology Strategy

The data in Figures 20 and 21, taken from an internal evaluation of Web Services technology at FS1, shows that technology managers are aware of the importance and benefits of using Web Services technology.

Technology managers realize that Web Services provides a new approach to application integration that is based on standard protocols and message formats. This overcomes a major limitation of the previous middleware standards, DCE and CORBA, which required applications to use the same software to integrate.

As the data show, managers recognize that it is strategically important to incorporate Web Services in the enterprise architecture and technology strategy.

Managers also realize that technology alone will not address their problems. FS3MD1 noted:

> I think this (Web Services) is a good approach. But we also realize that no technology will fix our internal issues. We now have several Six Sigma projects to address the bigger process issues. We are also working on our application architecture strategy.

Managers in this study indicated that the firms are now beginning to address their data and application integration problems by addressing the more fundamental issues of process and application architecture.

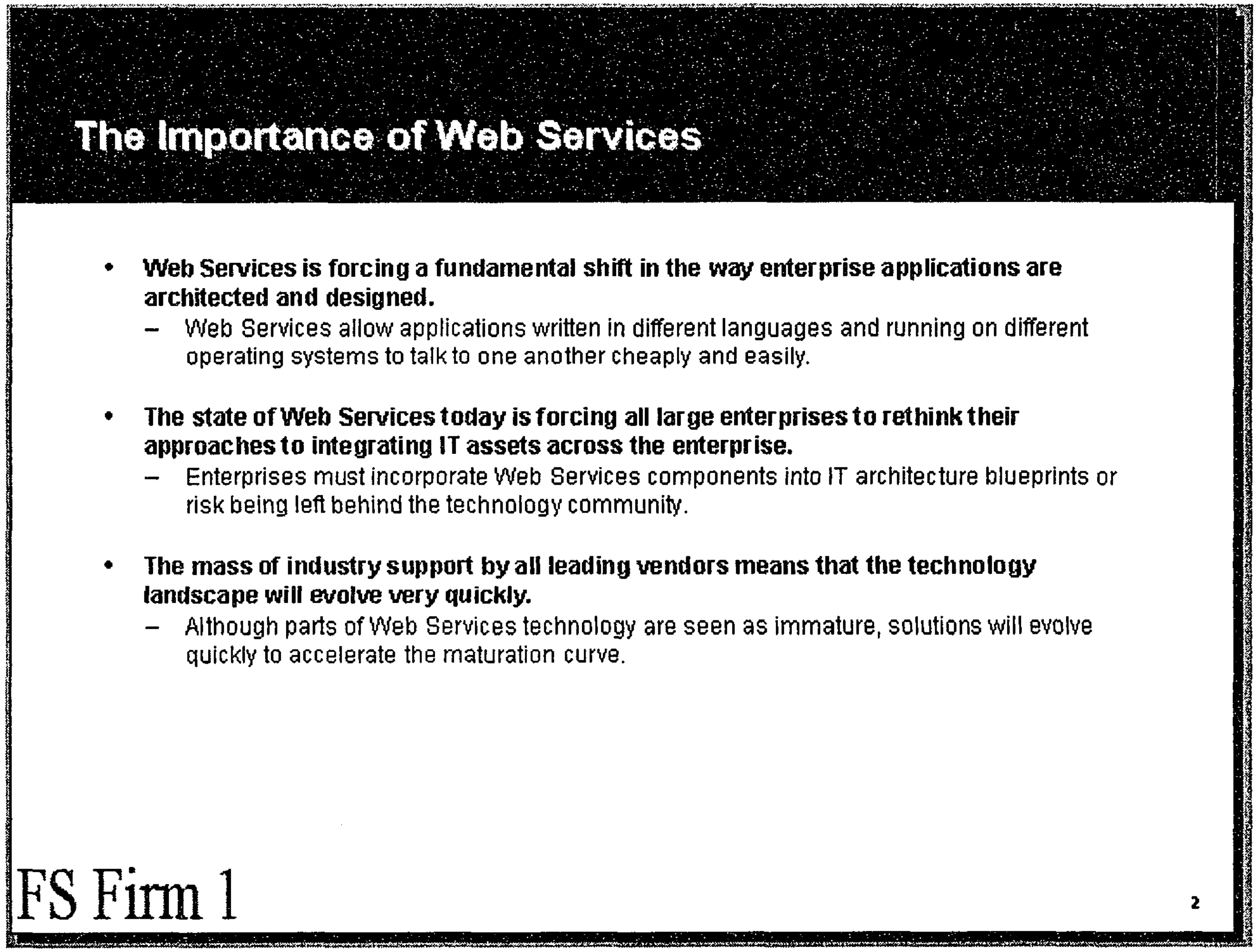

Figure 20: Importance of Web Services

Source: FS1 Internal Strategy Document dated February, 2003

Web Services: Benefits

- **The impact of Web Services is being felt across enterprises and will continue to grow in importance.** By adopting a Web Services strategy, JPMC can see a number of concrete benefits:
  - **Reduced costs due to simplified approaches to integration**, extended lifecycle of legacy systems, and use of standards-based technologies rather than proprietary ones
  - ***Increased competitive advantage and revenue*** *due to higher ability to deliver on core* business needs such as multichannel enablement, single view of customer, and private label partnerships
  - **Application Development benefits include:**
    - Modularity -- Web Services describe independent, self-contained business-level services
    - Interoperability -- Web Services allow interoperability across systems by using standard-based protocols and messaging formats
    - Accessibility -- Web Services provides the capability to publish individual components over public or private networks

FS Firm 1

3

Figure 21: Benefits of Web Services

Source: FS1 Internal Strategy Document dated February 2003

The following excerpt from and industry analyst report recommends a similar strategy (Gilpin, 2002):

> Those considering Web Services market viability should balance skepticism with cautious optimism that Web Services will ultimately deliver business value to many, if not all, organizations. In the meantime, early adopters should focus on a good business case and strong architectural approach to increase chances of success.

It is interesting to note that none of the user firms in this study are involved in the Web Services standardization process. This is in sharp contrast to DCE and CORBA where managers in these firms were very actively involved in shaping standards. FS2MD1 indicated that a lack of faith in the ability of these consortiums to deliver standards was one of the main reasons why he did not want to devote his time to these organizations. This was succinctly summarized by FS1VP1:

> Till the standards emerge, it's very risky to commit the firm to the technology. The vendors will have to cooperate, agree on standards, and develop products that we can use. And, this time, just an announcement in the press will not convince me! They (vendors) have done that too often in the past [i.e. announce that they agree to standards but do not develop products or continue to promote competing standards]. And finally, the businesses have to see real benefits before they will fund it.

The data indicates that technology managers are in the process of formulating their Web Services technology strategy and at the same time addressing business process and application architecture issues. They are cautious but optimistic about the future of Web Services but skeptical about the ability of vendors to cooperate.

The implementation of Web Services in organizations is discussed next.

Web Services Middleware Implementation

At this time, there are no widespread implementations of Web Services in the end-user firms studied as part of this research. The data from the consulting firms studied indicates that most organizations are either conducing pilots or are in the early stages of planning.

Since it will take a few years for organizations to deploy Web Services on a large scale, at this time there is not data available for "Web Services in use" and "consequences" categories.

This completes the description of the data for the implementation of Web Services middleware.

The next section provides data for the technology context that may impact the success of Web Services.

Web Services Technology Context

The data in this section corresponds to Theme III described previously and describes the technological factors, external to the organization, which influenced the implementation of Web Services middleware in organizations. As depicted in the middleware implementation framework in Figure 7, this theme is coded as the core category called technology context with the following sub-categories:

- Web Services technology standardization process
- Acceptance of Web Services standards by major vendors
- Availability of complementary products.

These sub-categories are described in detail in Table 16 and the following sections provide study data related to Web Services technology context organized in these sub-categories.

Web Services Technology Standardization Process

Study data indicates that the Web Services standardization process is an open process with the necessary feedback loops to ensure that standards are reviewed by industry experts, vendors and users. IA2SA1 noted:

> W3C has led the developmet of the highly successful web standards [in the past]. They are doing the same for Web Services. If you go through their documentation [at their web site], you'll find that the process is very open and anyone can comment on the specifications. Same for OASIS and WS-I. The problem I see is that there are these three cosortia involved in creating these standards and I'm afraind I still don't know who does what here!

The three consortiums that are involved in the development of Web Services standards are W3C, OASIS and WS-I. A review of their standardization process, based on the documentaton avaialblbe at their web sites, shows that they follow a very open process with provisions for review by industry experts, individual developers and vendors. This is in stark contrast to the DCE and CORBA standardization process where only participating vendors had the rights to review standards.

The confusion about the role of these consortiums was also reported in the following industry analyst report (Smith, 2003):

> All this has led to some instances of multiple proposed Web services standards, as well as competition among the standards bodies themselves — the Organization for the Advancement of Structured Information Standards (OASIS) and the World Wide Web Consortium (W3C). In many ways, this is not new. Having multiple standards and standards bodies is common. Web

> services had avoided such schisms and their effects, but the situation is changing. Both organizations claim they are not in competition, but we believe that they are, due in part to unclear differentiation. Although both have different internal workings (which are beyond the scope of this research), both claim to be standards-setting organizations for Web services and both accept specifications. It appears to be arbitrary or political motivations that are determining which (or both) body is involved with standardization efforts, rather than a clear area of ownership.

Similary, FS3MD1 noted:

> The success of Web Services will depend on the ability of these industry consortiums to resolve the conflicts quickly and produce something of value to the end-user. Otherwise, it will meet the same fate as DCE and CORBA.

CORBA and DCE standards were defined by a single consortium. In the case of Web Services, the confusion on the roles and responsibilities of the three consortiums may derail the standardization process.

Another important difference between Web Serices and the previous standardization attemtps is the tesing of standards as they are being defined. A survey of the web sites of three key vendors (IBM, Microsoft, and Sun Microsystems) and the three consortiums show that reference implementiaton of the standards are available for free download and testing. This ensures that the stndards are not developed in isolation but are being tested as they are developed.

The data show that the participants believed that the success of the Web Services standards will depend upon the cooperation among key vendors. IA2SA3 commented:

> As we have seen time and again, the success of these standards depends on the big vendors – if they buy in and produce products that work, these will go forward, else the [technology] community will move on to the next hyped thing.

This is discussed more fully in the next section.

Acceptance of Standards by Major Vendors

The study data indicates that all the key vendors have accepted Web Services standards. The following excerpt from an industry analyst report shows the unanimous agreement among vendors (Schulte, 2002):

> The unanimous vendor acceptance of Web services standards, especially the basic Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) specifications. Unlike CORBA and DCE, Web services standards have no naysayer among vendors. Rather, vendors invested heavily in Web services in 2001 and 2002, even before mainstream buyers were ready to use them.

As the data in Figure 22 indicates, vendors may resolve any conflicts in the Web Services standards purely out of self interest. The major vendors have different motives for accepting Web Services standards. For Microsoft, Web Services provides an alternative to the Java platform. For IBM, it is a differentiator from Sun Microsystems and also a platform to integrate its own myriad products and services.

This is unlike DCE and CORBA where some of the key vendors did not support the standards.

All the major vendors are investing in Web Services and developing products that meet the standards. This is discussed more fully in the next section.

Availability of Products

As the data in Figure 23 indicates, the vendors are already producing a plethora of products that are compliant with the Web Services standards.

"Web Services Will Succeed"

- Stone Soup
  - With everyone working on it, success is assured
  - It may take some time
- Convergent motives
  - For Microsoft, an alternative to Java everywhere
  - For IBM, a differentiator against Sun, etc.
  - For developers, blissful simplicity (perhaps)
    - RPC through firewalls (port 80)
    - "Test" by exposing services on the Internet

Figure 22: Vendors Accept Web Services Standards

Source: OMG Technical Committee Discussion (Welsh, 2002)

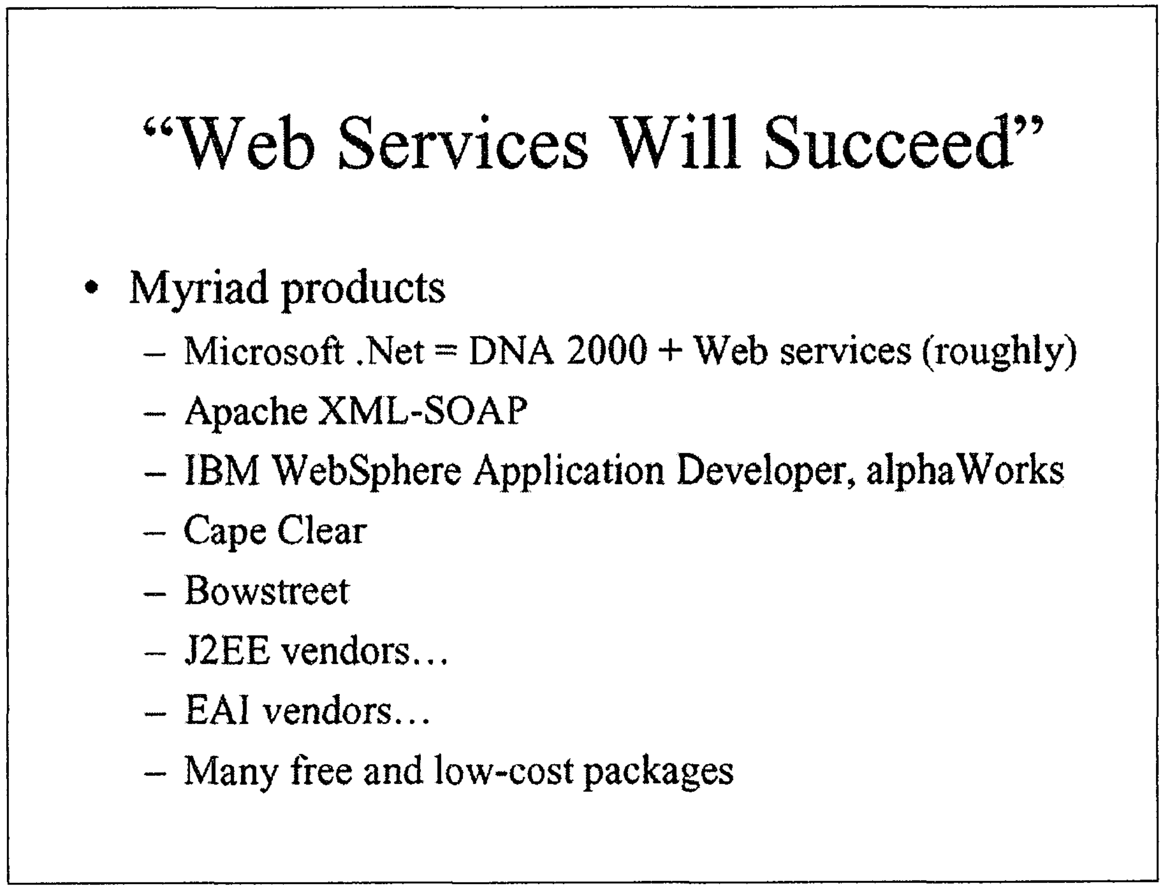

Figure 23: Myriad Products for Web Services

Source: OMG Technical Committee Discussion (Welsh, 2002)

Survey of the some of the key database and platform vendor's web sites including Oracle, Sybase, Sun Microsystems, IBM and Microsoft revealed that they all have Web-Services-compliant products.

Web Services is a key component of .Net product suite which is Microsoft's application development platform. Sun Microsystems has Web Services standards incorporated into their Java 2 Enterprise Edition application development product suite. IBM's Web Sphere product incorporates all the W3C recommended standards.

FS1VP1 noted:

> There are plenty of products available for download [over the Internet] from their (vendor's) web sites. This is what is different this time [compared to DCE and CORBA]. Then we couldn't even find a database vendor who was willing to work with us. Now every vendor who comes in here has a web services strategy.

The widespread adoption of Web Services standards by major vendors has created a viable market for complementary products.

This is in stark contrast with DCE and CORBA where few complementary products were available.

This completes the description of the data for Web Services technology context. The next section provides a brief summary of the study data for Web Services implementation process and the technology context.

Web Services Data Summary

The following quote aptly summarizes the state of the Web Services standards.

> At their very core, Web services are a continuation and evolution of enterprise middleware. Plenty of technologies have served as the 'prequel,' including the distributed computing environment (DCE), Common Object Request Broker

> Architecture (CORBA) Internet Inter-ORB Protocol (IIOP), Java Remote Method Invocation (RMI) and components. What sets Web services apart from these other, mostly failed technologies is the agreement of vendors on standards. The acceptance and endorsement of standards, in theory, delivers proliferation in the creation of Web services ...Proliferation of a technology drives the vendors to actually deliver on the hype; otherwise, the vendors' focus drifts to promoting the next great, hyped technology. (Lanowitz, 2003)

All the major vendors have accepted the Web Services standards. There are plenty of products available and these are being actively tested by users.

Technology managers are optimistic but cautious in their approach to Web Services. Organizations are incorporating Web Services as part of their architecture and strategy, but at the same time, are focusing on remedying their business processes and technical architecture. They realize that technology products alone will not solve their data and application integration issues.

This concludes the description of the data for the implementation process and technology context for Web Services.

SUMMARY OF FINDINGS

The data indicates that the failure of the last two generations of middleware standards, DCE and CORBA, was primarily due to three interrelated issues.

First, the standards failed to meet the needs of the end users. This was partly due to the limitations of the process that was used by industry consortiums to create the standards. Both OSF and OMG were closed to the broad technology community and served only the interests of the participating vendors. As such, the standards were developed in isolation without the experience and expertise of the technology experts who could have ensured that the standards met the needs of end users.

Second, DCE and CORBA were not adopted by some of the key vendors who promoted their own products and standards. This resulted in competing standards and prevented the wide-spread use of DCE and CORBA. When competing products are available from a major platform vendor, end-users are hesitant to adopt a technology that may eventually be replaced by the stronger player.

The third factor that limited the adoption of DCE and CORBA was a lack of complementary products. The producers of complementary products like databases and firewalls did not adopt DCE and CORBA standards. Since these products are widely used in most large organizations, a lack of integration with them prevented wide-spread use of these standards in organizations because they did not fit in with products already in use.

The general consensus among the participants was that the success of Web Services will depend upon avoiding these pitfalls.

These findings suggest that technological factors had more influence on the success on middleware technologies than the organizational ones.

Why did vendors fail to produce middleware products that could meet user's needs? Why didn't some of the major vendors cooperate to make the CORBA and DCE standards successful? What are the factors that influence a vendor's decision to support or reject a technology standard? What are the factors that influence the production of complementary products by vendors?

These are some key questions that can provide us more insight into the causes of the failure of the last two generation of middleware technology. These can also help us understand the forces that will shape the success of Web Services. The next chapter discusses existing research that provides more insight into these questions and provides a deeper understanding of forces that influence the success of middleware technology.

CHAPTER 5

DISCUSSION AND IMPLICATIONS

In this chapter, the findings of this study are interpreted in the context of an existing body of theory from the technology innovation research area. This is a common practice in grounded theory research where an existing literature can "be used to confirm findings" and also for "extending, validating, and refining knowledge in the field" (Strauss & Corbin, 1998, p. 49).

The concept of a dominant design from the technology innovation literature provides insight and also better understanding of the factors that affect the success of new and competing technologies. It has direct application to middleware technologies and will be the main concern of this chapter.

The empirical model developed in Chapter 4 identifies some of the key factors relevant to the success of middleware technology. In this chapter, these specific empirical findings are integrated into the broader theoretical framework of dominant design with the aim of enhancing an understanding of the forces that impact the success of middleware technology. Specifically, findings from the existing literature are used to provide another level of understanding and insight to the answers that grow from the following questions:

- Why did the vendors fail to produce middleware products that meet the needs of users?

- What are the factors that influence a vendor's decision to cooperate or compete with new or existing standards?
- What are the factors that influence a vendor's decision to produce complementary goods for a particular technology?

This analysis is then used to derive recommendations for practitioners and researchers.

This chapter is structured as follows. First, relevant ideas from the technology innovation literature are discussed with a focus on the concept of "dominant design" and its relevance to middleware technology. Then, the factors that impact the success of middleware technology are analyzed in light of this existing body of theory. Finally, recommendations for researchers and practitioners that emerge from this analysis are discussed.

DOMINANT DESIGN AND MIDDLEWARE

As depicted in Figure 24, technology change at an industry level has frequently been characterized as a cyclical process. Researchers have identified two events that trigger changes in the industry structure: technological innovation and emergence of dominant design. These two events that trigger changes in the industry structure, also change the innovative focus of firms that produce the technology, as well as rules of engagement in the competitive war for dominance in an industry (Anderson and Tushman, 1990; Tushman and Anderson, 1986; Utterback and Abernathy, 1975).

New and innovative technology triggers an "era of ferment." During this period, the industry is marked by multiple competing technologies, lack of standards, and frequent technology changes. In an effort to grab market share, firms producing the

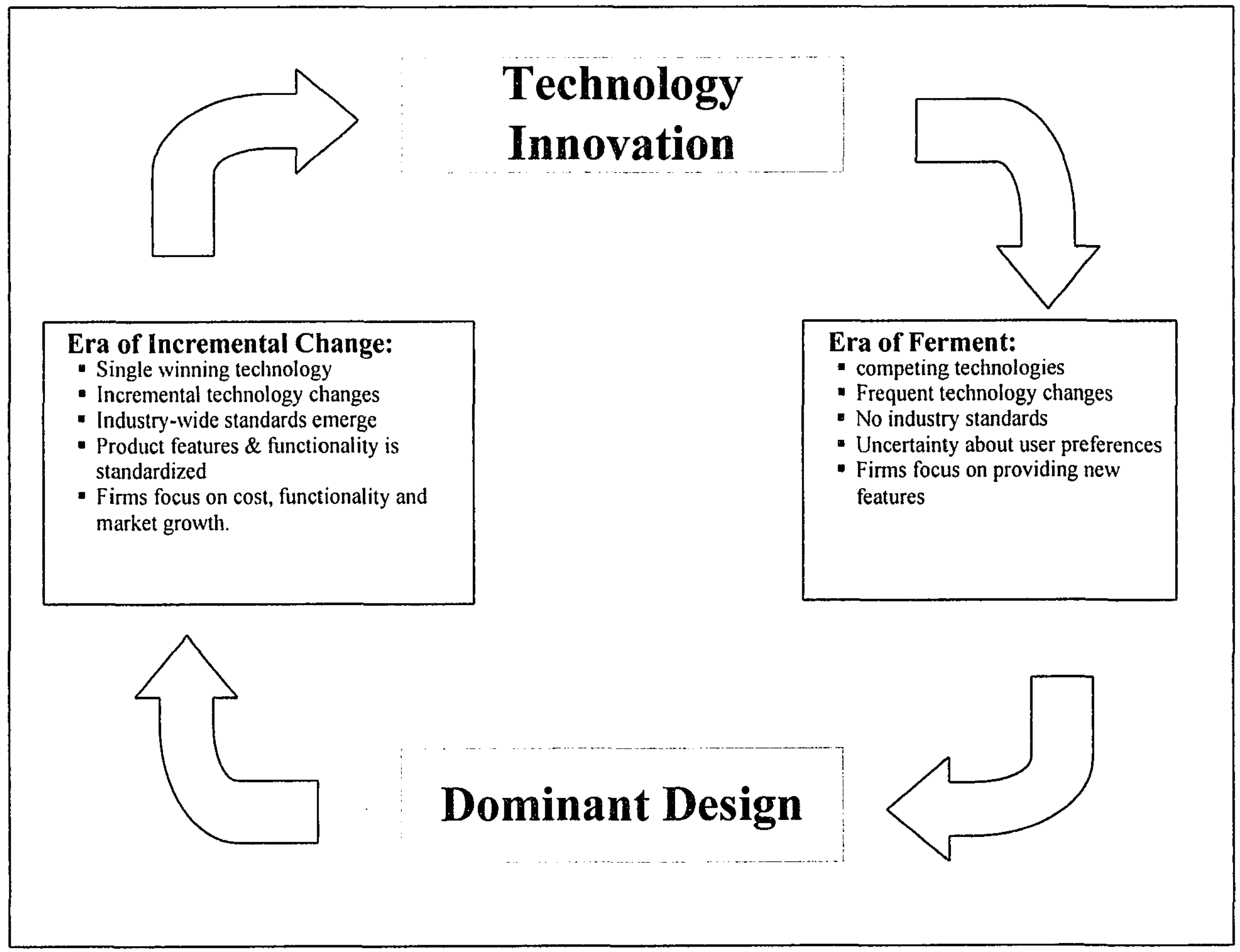

Figure 24: Technology Evolution Model

Source: Adapted from Tushman & Anderson, 1986; Tushman & Rosenkopf, 1992; and Utterback & Abernathy, 1975 technology focus on providing new features in products as the preferences of users are not clear.

Forces, both technological and non-technological, may lead to the establishment of a particular technology as the dominant one and result in the elimination of competing technologies. This winning technology has been termed a "dominant design" by Utterback and Abernathy (1975).

From the data discussed in Chapter 4, it is evident that the middleware industry is in an "era of ferment" with no standards, frequent technology changes and no clear dominant technology. The two previous attempts by industry consortiums to create middleware standards, DCE and CORBA, failed to gain acceptance in the market. Web Services is the third attempt by the industry in the last 14 years to create a middleware standard. Past research predicts that these technology changes in the middleware industry will continue until a dominant design emerges and the technology stabilizes (Utterback & Abernathy, 1975).

The emergence of a dominant design "marks a significant watershed in the competitive nature of an industry" (Christensen et al, 1998, p. 208) and has significant implications for both producers and consumers of technology.

Technology producers whose design is not the dominant one can be eliminated from the competition (Christensen et al, 1998; Suarez & Utterback 1995, Schilling 1998). Consumers who bet on the losing technology are locked into a technology that becomes increasingly more difficult to support as their suppliers are eliminated, complementary goods and services are not available, and supply of trained resources dwindle. They are ultimately forced to migrate to the dominant technology or keep incurring increasing costs to support the legacy technology if high switching costs prevent the migration (Schilling, 1998; Lieberman and Montgomery, 1988).

This has been the experience of the DCE and CORBA users in the firms studied in this research project. As the vendors stopped providing DCE support, users were forced to migrate their applications and infrastructure to other platforms. The current users of CORBA are in the process of migrating to other technologies or waiting for Web Services standards to emerge before making a technology choice.

The emergence of a dominant design triggers the era of incremental change during which alternative designs are driven out of the market and the focus of innovation shifts to production efficiency, incremental changes to the dominant design and market growth (Tushman & Rosenkopf, 1992). "For example, while the basic technology underlying xerography has not changed since Carlson's Modle 914, the cumulative effect of numerous incremental changes on this dominant design has dramatically improved speed, quality, and cost per unit to reprographic products (Dessauer, 1975)" (Tushman & Anderson, 1986, p. 441). "These incremental technological improvements enhance and extend the dominant technology and thus reinforce an established technical order" (Tushman & Anderson, 1986, p.441). This process continues until a new breakthrough occurs and the entire cycle is repeated.

The emergence of a dominant design reduces consumer confusion, uncertainty and the risk of adopting the new the technology because it reduces variation among competing products (Abernathy & Utterback, 1978). This also promotes compatibility and integration with other products including complementary products (Farrell & Saloner, 1985). The overall affect is an increase in demand that promotes economies of scale and a learning curve affect for both consumers and producers (Schilling, 1998).

The concept of dominant design is strongly linked to the notion of product standards. Various authors use the terms "dominant design", "product standard" and "dominant standards" interchangeably (Schilling 1998). Suarez and Utterback (1995) state that the notion of a dominant design is broader than that of standards and includes standards. In the following discussion, these terms are used interchangeably.

Given the impact of dominant design on both the producers and consumers of technology products, researchers have studied the factors that determine whether a particular technology will become dominant in the market or not. These findings provide further insight into the factors that impact the success of middleware technologies.

The next section describes the factors that influence the emergence of a dominant design in an industry and discusses those that apply to middleware in detail.

FACTORS AFFECTING DOMINANT DESIGN IN MIDDLEWARE

The process by which a technology becomes dominant in the market is not well understood. Suarez and Utterback (1995, p. 416) postulate that "the emergence of a dominant design is the result of a fortunate combination of technological, economic, and organizational factors." Lee et al. (1995) describe the process as a "black box" with various factors that are difficult to identify and measure.

Extensive research has clearly established that technological superiority is the not the main driver behind the emergence of a dominant design (Suarez & Utterback, 1995;

Teece, 1986; Tushman & Rosenkopf, 1992). Table 18 summarizes the factors that have been identified by researchers as relevant to the emergence of a dominant design.

As this summary indicates, a vast array of technological, economic, organizational, and governmental factors can impact the success of a technology in the market. For example, network externalities heavily influence the emergence of a dominant design in industries involving physical networks like telecommunications, railroads and faxes (Schilling, 1998). Network externalities, availability of complementary goods, and learning curve effects can heavily influence success in the software industry (Farrell and Saloner, 1986; Schilling 1998, 1999)

Since the specific factors and the degree to which they affect the emergence of a dominant design vary from industry to industry, it is necessary to develop an industry specific framework that can be used to analyze and understand the factors that lead to dominant technologies in specific industry sectors (Lee et al., 1995).The findings of this research, discussed in Chapter 4, indicate that the following factors impacted the success of middleware technologies:

- Ability of middleware technology to meet the needs of users
- Support of middleware standards by key vendors
- Availability of complementary products.

The following sections discuss insights from the dominant design literature as they apply to these middleware industry-specific factors.

Table 18: Factors Affecting Emergence of a Dominant Design

Factors	Description
Network Externalities	This refers to the phenomenon where the benefits a consumer derives from the technology or a product increases as the number of other consumers using it increases (Katz & Shapiro, 1986, 1992, 1994). For example, owners of fax machines found them more useful as more users bought compatible fax machines.
Learning Curve Effects	This refers to the phenomenon where the more a technology is used, the further it is developed and made more effective and efficient (Schilling, 1998).
Complementary goods	Many products have no value in isolation but are useful only when used with other (Katz and Shapiro, 1994). Such products are called complementary products. The greater the availability of complementary goods for a particular technology, the greater are its chances of being established as a dominant technology in the market place (Schilling, 1998).
Demand-Side Forces	These are the factors that affect consumer demand and in turn can affect the emergence of a dominant design. The following two factors affect consumer demand: (1) Risk avoidance induces potential buyers to delay use of a new technology until a dominant design emerges (Bensen and Farrell, 1994). (2) The emergence of a dominant design promotes increase in demand and this, in turn, lowers costs as economies of scale kick in (Suarez and Utterback, 1995).
Supply-Side Forces	These are the forces that promote suppliers of technology to converge towards a dominant design. Efficiency and collusion are two supply-side forces that have been identified by researchers (Lee et al., 1995). The following factors have been identified in previous research: (1) The emergence of a dominant design increases efficiency because it reduces the number of alternate designs. The marketing costs are also reduced because of simplified consumer communications. (2) Since the emergence of a dominant design reduces the number of alternatives a consumer has, it may lead to tacit collusion among vendors.

Factors	Description
Strategic Maneuvering at the Firm Level	This refers to the type of strategy followed by a firm with respect to its products as compared to its competitors. It may determine which firm's product becomes dominant (Suarez and Utterback 1995). For example, Matsushita effectively partnered with a number of vendors to promote its VHS standards and managed to significantly increase its installed base and the availability of complementary products. Sony, on the other hand, believed in the technological superiority of its Betamax standards and did not provide other vendors rights to its technology. Eventually, VHS became the dominant technology in the market because of Matsushita's product strategy.
Appropriability	This refers to environmental factors, other than firm and market structure, that impact the ability of a technology producer to capture the profits produced by the innovation (Teece 1986). For example, patents can prevent other vendors from entering the market. The products from the owner of the patent can become dominant in the market. For example, Xerox owned a number of patents for photocopy technology and their products dominated the market until the patents expired.
Government Regulation	In some industries, like telecommunications and utilities, government mandated regulation can induce adherence to a dominant design (Anderson & Tushman, 1990). For example, FCC's approval of RCA's television broadcast standard established its design in the television industry (Suarez and Utterback, 1995).
Technological Factors	These are characteristics of the technology that may determine its success in the market. The following factors have been identified by researchers (Lee et al., 1995): (1) Technical feasibility of the technology determines if it can be converted into a viable product that is available to users at a reasonable cost. (2) Degree of fit of a new product or technology with existing technologies can also impact its adoption.

Ability to Meet User Requirements

As the findings of this research indicate, DCE and CORBA failed to meet the needs of users. Why do firms that produce technology products fail to meet the needs of the market? Past research can provide insights to this question along the following two dimensions:

- nature of the innovation process
- characteristics of the standardization process.

Often with new technology, there is little or no consensus on what user's needs are. In this case, the producers of technology take a gamble on what they believe are the consumer's needs (Schilling 1998). The vendors then iteratively evolve the technology based on perceived needs and feedback from users. Over time, as the technology evolves and customers requirements are better understood, a dominant design may emerge that aggregates user needs (Tervolin, 2000).

The findings of this research indicate that the standardization process that led to the creation of DCE and CORBA standards was not optimal. With DCE middleware technology, the standards were accepted prematurely. There was no concerted effort to understand the needs of users and evolve the technology in response to user feedback. This led to a product that could not meet user needs. In the case of CORBA, the technology evolved iteratively over time, but it took too long for it to mature to a point where it could meet user needs.

Existing research identifies the characteristics of the standardization process that can positively influence the outcome. Rutkowski (1995), from his analysis of the success of the Internet standards process, identifies the following critical success factors:

- an emphasis on meeting user needs
- an outcome of the process which is a standard that works in the field
- active and open participation by individuals and industry experts
- a comprehensive and thorough review process to ensure quality.

The data shows clearly that the standardization process for DCE and CORBA lacked these critical features. For example, only OSF and OMG members were allowed to participate in the creation of the DCE and CORBA standards. Industry experts and individual users could not contribute to these standards. The technologies that were incorporated in the DCE standards were not tested in the field and the participants in the process were mostly engineers and technologists who had little experience in the business world where these products were going to be used.

The process for the standardization of Web Services appears to have all these characteristics.

Sherif (2001) identifies three kinds of standards that result from the different processes:

- Responsive standards
- Participatory standards
- Anticipatory standards.

Responsive standards are created when a successful product is available in the market. A dominant product becomes a de facto standard in the market.

Anticipatory standards are created without having an actual working product. These are created in response to anticipated or perceived needs. DCE and CORBA standards were anticipatory standards.

Participatory standards are created in parallel with product development. Here, as the product is developed and used, the feedback is used to refine and change the standards. The Web Services standardization process can be characterized as participatory.

Sherif (2001) further concludes that participatory standards are optimal when the technology is rapidly gaining acceptance in the market. Anticipatory standards require active participation by all stakeholders or else they can fail.

In the case of the DCE and CORBA standardization processes, some of the key vendors did not participate in the process. The reasons for this are explored further in the next section.

Support of Middleware Standards by Key Vendors

The data shows that DCE and CORBA standards failed to get established in the market because they did not have the support of some key vendors.

What factors influence a vendor's decision to cooperate or compete with its rivals over standards?

The firm whose technology becomes dominant in the market will have an extremely profitable position in the market. Examples of this include IBM's dominance in the mainframe market and of Microsoft's dominance in the desktop market (Bensen and Farrell, 1994). Since the rewards of owning a dominant technology are so lucrative, firms compete fiercely to ensure that their technology is selected as the dominant one in the market. But competition can also drain profits and this acts as restrainer from engaging in an all-out standards war with well established rivals (Bensen and Farrell, 1994).

A decision to adopt a standard can increase the profitability for some vendors and reduce it for others (Katz and Shapiro, 1985). A firm's strategy towards standardization depends on both its view about how likely it is to prevail in the standards battle and also on the expected payoff of standardization (Bensen and Farrell, 1994).

If vendors are well established in the market and have a large installed base of their products, then there is little incentive for them to support an industry standard that would allow rival products to substitute their own and reduce their profits (Katz and Shapiro, 1985). The findings of the present research substantiate this assertion. For example, when DCE was introduced, both Sun Microsystems and Microsoft were well-established vendors with competing products and so they had no incentive to support the standards effort because it would have eroded their installed base and their profits. Similarly, there was no incentive for Microsoft to support CORBA because it had a competing product, COM, which could have emerged as a dominant product in the market.

Bensen and Farrell (1994) identify three strategies vendors can adopt towards standardization:

- Standards Battle: Firms decide to engage in an all-out battle to establish their standard as the dominant one. Firms typically choose this strategy when they are on equal footing with their rivals and the unavailability of standards does not impact adoption by users.
- Standard Negotiation: Firms adopt this strategy when compatibility is important to generate consumer demand. In this case, each player prefers a standard to no standard, but each prefers its own standard to other's. In such a scenario, it is very

common to see companies continuing to develop their own proprietary solutions, even while engaged in standards negotiation.

- Standard Leader: This strategy is adopted by dominant players when the rivals are not well established. In this case, a dominant player attempts to prevent the weaker players from adopting their technology as the standard. They do this either by enforcing intellectual rights to the technology or by changing their technology frequently so that followers are always behind.

Clearly, in the case of DCE and CORBA standards, the established vendors adopted the standards battle strategy. The end result was that none of the vendors gained dominance in the market. Since there is no single dominant vendor in the middleware industry at this time, the standards leader strategy is not a viable option.

Since middleware is used to integrate the myriad systems produced by different vendors, unless the technology is supported by all major vendors, it will not serve its primary use – integration of diverse platforms. From this perspective, standards negotiation appears to be the right strategy for middleware vendors.

The data shows that for Web Services standards, all the key vendors are involved in the standards negotiation process. So there is a very strong possibility that Web Services may emerge as the middleware standard in the near future.

Vendors prefer to agree on standards when the lack of standards makes the demand weak or when a standards battle will dissipate much of the potential profits or if agreeing on standards will give the company an advantage over its rivals (Bensen and Farrell, 1994). When there is a possibility that adopting standards may increase the total value of the industry so much that it overcomes the dilution in profits, then dominant players in an industry are open to adoption of standards (Shapiro and Varian, 1999). This may be another reason that is prompting vendors to cooperate to create Web Services standards.

Uncertainty over standards can also encourage buyers to adopt a "wait and see" attitude before they make a technology choice to prevent lock-in into a losing technology (Bensen and Farrell, 1994). The results of this study clearly indicate that after the failure of DCE and CORBA standards, users were and still are reluctant to adopt new middleware technology on a large scale. The possibility of a significant increase in market size may be another reason that is driving middleware vendors to cooperate to create new middleware standards.

The discussion in this section demonstrates that the behavior of vendors and users in the middleware technology industry closely conforms to the theoretical findings from the dominant design literature.

The next section discusses the factors that impact the availability of complementary products.

Availability of Complementary Products

"Many technologies are not useful or desirable to customers without an associated set of complementary goods (e.g., computers and software, VCRs and videotapes, or copy machines and toner cartridges) (Schilling, 1998, p. 275)."

Research shows that in markets characterized by network externalities, the availability of complementary goods can impact the success of competing technologies (Farrell and Saloner, 1985; Katz and Shapiro, 1986, 1992). Network externality is a phenomenon where the benefit a user derives from a product increases as more and more users use the product (Katz and Shapiro, 1986). In information technology related industries, availability of complementary products and network externalities play a more significant role than other factors (Bensen and Farrell, 1994).

As the findings of this research shows, incompatibility of DCE and CORBA with databases and firewalls limited their usefulness in organizations. Why didn't the database and firewall vendors integrate their products with DCE and CORBA? Past research may provide some insights into this question.

As depicted in Figure 25, there is a self-reinforcing affect between the availability of complementary goods and the dominance of a technology in the market. Since the cost of supporting multiple technologies is high, the manufacturers of complementary goods prefer to produce goods for a technology with a high number of users. The availability of complementary goods for a particular technology, in turn, prompts users to select that technology over competing ones which increases its installed base (Schilling 1998).

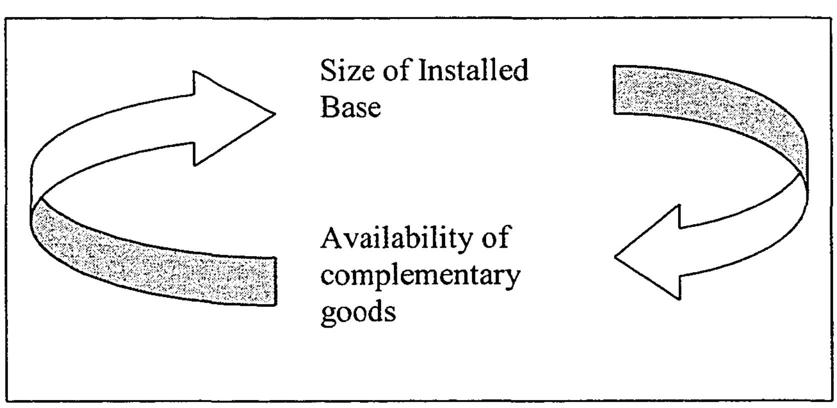

Figure 25: Installed base and complementary goods reinforce each other
Source: Schilling (1999)

Since the installed base for DCE or CORBA was not high, the database manufacturers did not find it profitable to integrate their products with these technologies. For consumers, lack of database integration, in turn, prevented them from using DCE and CORBA which reduced their installed base. It can be postulated that the success of Web Services will depend on the adoption of Web Services standards by the database and firewalls vendors.

The conclusion from this discussion is that forces that shape the emergence of a dominant design in the middleware industry are similar to those in other industries. The implications of this conclusion are discussed in the next section.

RESEARCH IMPLICATIONS

The theoretical framework developed in this research identifies the factors that impact the successful implementation of middleware in large global organizations. This research indicates that the forces that influence the success of middleware technology are the same as those that impact the success of other technologies and have been investigated in relation to the concept of dominant design. The findings and the framework developed here have implications for both the research and practice of technology innovation and implementation. These are summarized in Table 19.

Implications for Researchers

This research adds an understanding of the forces contributing to the success and failure of information technologies in end-user firms.

The existing frameworks and models that focus on information technology implementation share four characteristics:

- They identify stages of technology implementation in an organization (Cooper and Zmud, 1990; Preece 1991; Chengalur-Smith & Duchessi 1999).

Table 19: Research Implications

Constituents	Implications
Researchers	▪ Existing frameworks and models for technology implementation must be extended to include technology evolution as a critical factor that impacts success. ▪ The dominant design framework from technology innovation literature can also be applied to middleware and other IT products to evaluate the maturity of a particular technology. ▪ Researchers should attempt to identify the strategic implications of using an evolving technology and identify conditions that may help managers decide when to use it. ▪ No dominant design has yet emerged in the middleware industry. This is an opportunity for researchers to study vendor strategies, standardization processes, and the economic and competitive forces that determine the emergence of a dominant design in middleware industry.
Middleware Technology Managers	▪ Technology managers should take into consideration the maturity of technology to avoid possibility of a lock-in into an unstable technology that may lead to expensive migrations. ▪ They should design solutions that are, where possible, product independent so that if the products change, there is minimal impact on the systems. ▪ Managers should monitor middleware vendor strategies and participate in the middleware standardization process to improve their ability to evaluate the maturity of technology products they are planning to use.
Middleware Technology Vendors	▪ Vendor should design products with integration and interoperability in mind as these are the primary features that users value. ▪ Integration with products already in use at firms, particularly databases and firewalls, is critical for the success of middleware products. ▪ Standards should be developed via an open process that encourages participation from a broad category of constituents and includes vigorous independent reviews and field testing. This improves the quality of the software and its ability to meet the users needs. ▪ All the dominant vendors should participate in the standardization process to ensure its success.

- They address the impact of contextual factors on various stages of implementation (Kwon & Zmud, 1987) ;
- They analyze the characteristics of the technology to identify features that impact its adoption and use (Rogers, 1983); and
- They ascertain the impact of user's perception of technology on its successful implementation and use (Davis, 1989).

These models and frameworks have been documented in detail in Chapter 2. These existing models explain the results of technology implementation for technologies that are stable and do not change frequently, for example databases and CASE tools. But they fail to explain the issues related to implementation of technologies like middleware that are still evolving.

Figure 26 depicts a more comprehensive technology adoption and implementation model that is a synthesis of previous research and the findings of this research. This model accounts for the various factors that impact the successful implementation of information technology in organizations. These include:

- identification of the stages of the adoption and implementation process;
- affect of the contextual factors that impact the various stages of the process including characteristics of the technology, organization, users, fit between the task and the technology used to perform the task, and the environment; and
- affect of the technology's evolution along the innovation cycle that leads to a "dominant design" and its impact on the contextual factors within the various implementation stages.

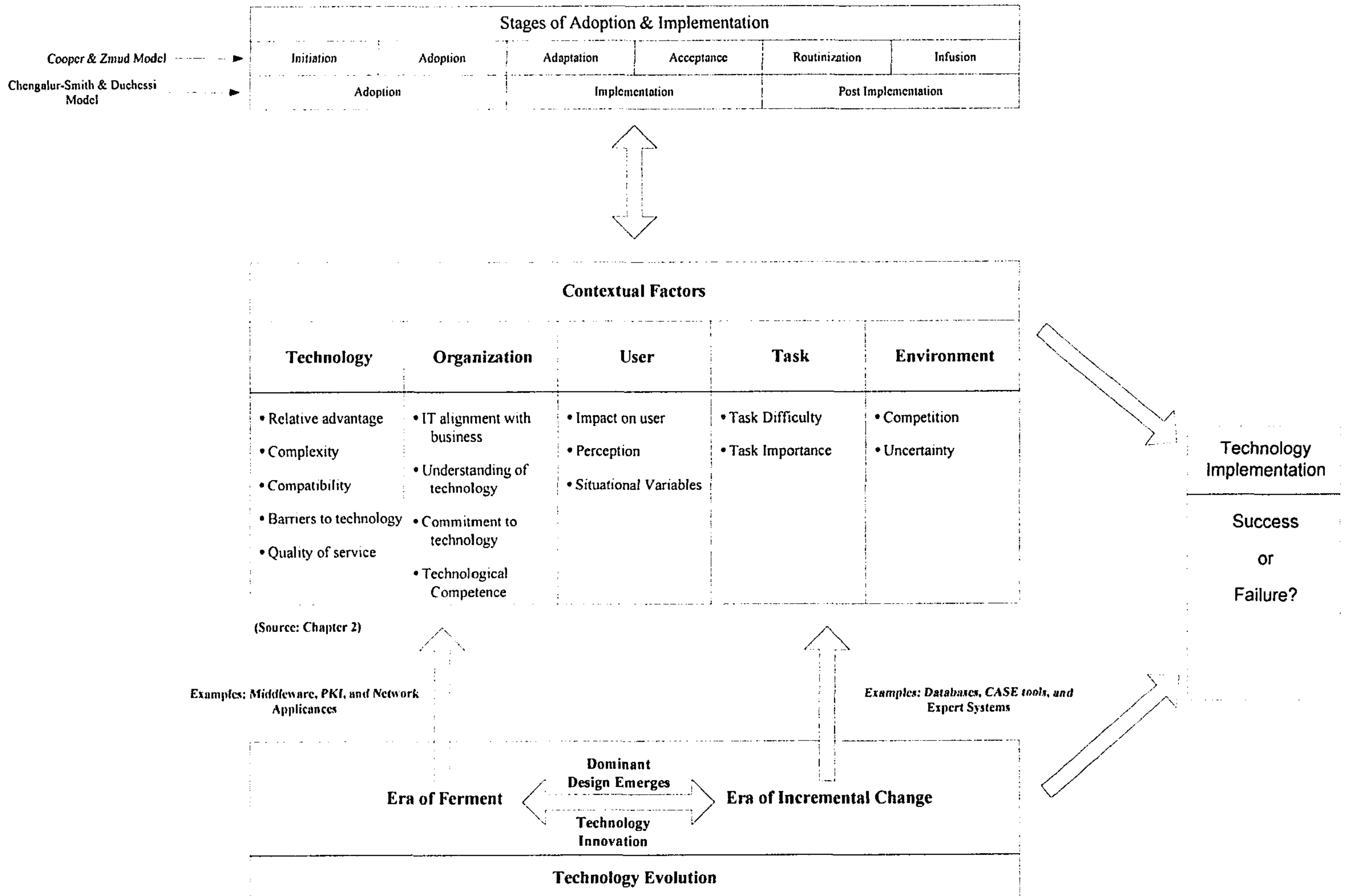

Figure 26: New Technology Adoption and Implementation Model

Past research, documented in detail in Chapter 2, has clearly established the impact of these contextual factors on the various stages of technology adoption and implementation. It also shows how the affect of these factors depends on the technology under consideration. For example, for Expert Systems' success in organizations, user's perceptions about its negative impact on their job security is more significant than the perceived benefits they derive from it (Guimaraes et al., 1996). For CASE tools, perceived benefits of using the technology are more significant (Chau, 1996). These factors are retained in this new model.

This model incorporates the findings of this research by adding the affect of the state of the technology along its evolution path on the success or failure of its implementation. As the figure depicts, the state of the technology is conceptualized by "era of ferment" and "era of incremental change." The emergence of a dominant design and new technological breakthroughs are the two key events that cause the transitions between these two stages. For a specific technology, its state impacts the contextual factors and also the success of implementation of that particular technology.

This model postulates that the concept of dominant design can provide a relevant framework to identify and evaluate the maturity of technology under consideration for implementation and use. It indicates that products will keep changing until a dominant design emerges and technology mangers will have to account for these changes in their planning and design.

Frequent changes in technology create conditions that can have an adverse affect on implementation projects. For example, in typical large global organizations with thousands of applications, it may take one to two years to deploy a new version of

Microsoft's desktop operating system. During this time period, Microsoft would have produced at least one more version of the product and so by the time the deployment is complete, the product that is being used is already behind the current vendor versions. Understanding the impact of such evolutions becomes even more critical when the newer technology completely obsoletes the older ones.

More empirical validation and elaboration of these concepts in the context of different technologies are clearly needed. In the information technology industry, there are several technologies that are in a state of ferment where a clear winning dominant design has not yet emerged. For example, Public Key Infrastructure (PKI) technology that is used for electronic signing of documents, authentication, data privacy, and data protection has not yet matured and there are several cases of failed deployment in the industry. Further studies of such technologies may identify other forces that can influence the dominant design theory and add to our understanding of how these forces impact the adoption and use of such systems in organizations.

Additionally, researchers should attempt to develop the strategic implications of using technology that is inherently unstable because no dominant design has yet emerged. Should end-user firms wait for a technology to stabilize before they adopt it on a large scale? Will they forgo any strategic advantage if they adopt a wait-and-see strategy? Under what conditions should firms adopt new but unstable technology? Researchers should aim to develop strategic frameworks that can answer some of these questions and test these ideas empirically.

There is a clear need for more research in contemporary industries to identify factors that influence the establishment of a dominant design (Suarez and Utterback, 1995). This research provides an understanding of some of the factors that will impact the success of middleware standards. However, the focus of analysis in this research was the end-user organization. Since a dominant design has not yet emerged in the middleware industry, this is an opportunity to study vendor strategies and actions to further increase our understanding of the following:

- Why are the major middleware vendors (IBM, Sun Microsystems and Microsoft) cooperating to produce the Web Services standards?
- What will be the impact of standardization on the vendors? Which ones will win and which ones will lose and why?
- Why did Microsoft and Sun Microsystems not cooperate to produce the DCE and CORBA standards?

In addition to vendor strategies, future research could attempt to identify the middleware technology characteristics along with economic and competitive forces that are shaping the cooperation among major vendors. This investigation can provide further validation and insight into the models developed in diverse streams of literature including strategic management and economics (Bensen and Farrell, 1994; Shapiro and Varian 1999).

The technology standardization process is another area of research that can benefit immensely from further investigation, especially as it concerns the Web Services standardization process. The three key standard bodies (W3C, WS-I and OASIS) are working together to create the Web Services standards and have an open process. This is an excellent opportunity to observe the standardization process where vendors are cooperating and can provide empirical evidence of the models suggested by Bensen and Farrell (1994).

Implications for Practitioners

This research has important implications for practitioners, both technology users and vendors.

This research demonstrates that users of technology should take into account the state of technology standards development when evolving their technology strategy. This will help avoid the possibility of technology lock-in or expensive migrations if the technology they select does not become dominant in the market. Managers should weigh the advantages of using a new technology that may change in the near future against the costs involved in migrations and retraining.

End users should consider the possibility that the technology they are selecting may not become dominant in the market. Where possible, they should design a solution in such a way that it is independent of the product being used. This will allow a plug-and-play solution to be developed so that the underlying technology can be replaced with less pain. This is even more important for middleware technologies that are deployed as an infrastructure because of long gestation periods and the large number of applications that depend on it.

This research provides insight into the factors that impact the emergence of dominant design and standards in middleware technology and can help managers make an early determination of which designs have a higher probability of becoming dominant in the market.

Technology managers should monitor the middleware industry and pay special attention to vendor strategies and the standardization process. This will improve their ability to evaluate the stages of the technology evolution of the products they are selecting for firm-wide deployment.

This research indicates that middleware technology standards that are likely to dominate the market will have the following characteristics:

- are accepted by the major platform vendors
- integrate with complementary products like databases and firewalls
- are protocol based.

This will also help technology managers make technology bets with more confidence and account for predictable changes in their deployment strategy and in the design of their infrastructure. This also helps in internal communications with the stakeholders and facilitates setting expectations for the technology, particularly returns on investment and life of the infrastructure

Technology vendors in the middleware space should consider that technology developed with interoperability and compatibility with existing products in mind has higher value to end-users and has a better chance of becoming dominant in the market. This research indicates that for middleware technology, integration with databases and firewalls is essential for adoption in large global corporations which use middleware as an integration tool.

Ambiguity about user requirements can very often lead to failure of technologies (Schilling, 1998). This research indicates that one key requirement is the ability of users to integrate systems without mandating what technology their partners use. This indicates that the middleware standards will have to be based on standard protocols and not dictate what product to use.

This research indicates that the process by which middleware technology is standardized influences its quality and ability to meet the needs of users. In the complex arena of middleware technology, it is important to have an open process that encourages participation from a broad category of constituents and includes vigorous independent reviews and field testing.

The vendors engaged in the standardization of middleware should consider the importance of having all major vendors involved in the process. Since middleware is used primarily for integrating diverse platforms, unless the vendors who provide the dominant platforms in the industry agree to the standards, the standards will not find widespread use.

LIMITATIONS OF THIS STUDY

This study is based on extensive interviews with:

- key end-users of middleware technology,
- consulting firms that bridge the gap between providers of the technology and consumers, and who recommend appropriate technology solutions to the consumers;,and
- industry analysts that analyze and predict the future of the technologies.

It does not take into account industry-specific data that has been done in previous studies related to the dominant design concept. For example, Christensen et al. (1998)

studied detailed industry reports on disk drive technology including sales figures and prevalent designs.

In the middleware technology arena, such data is not readily available and most companies do not report middleware specific figures – they bundle it with their overall software sales figures. Since the middleware industry is still evolving, this is a window of opportunity for researchers to track data on individual vendors and use it to further develop the findings of this research.

CONCLUSIONS

The findings of this research reveal that the lack of adoption of middleware technologies by users has more to do with broader industry-wide technological factors than with internal organizational ones. Specifically, a dominant design has not yet emerged in the middleware arena and that has led to frequent technology changes which have negatively impacted users.

This research has identified the key factors that have influenced the failure of previous generations of middleware standards and will impact the success of current industry efforts to standardize Web Services as the new middleware standard emerges.

Suggestions for future research have been detailed and these will further our understanding of the forces that impact the success of middleware technology.

Implications of these findings for theoreticians and practitioners have been detailed. Guidelines have been provided for practitioners, both users and vendors, so that they can utilize the findings of this research in their day-to-day activities.

APPENDIX A

MIDDLEWARE INDUSTRY ACRONYMS

ACRONYM	**Description**
BPEL4WS	Business Process Execution Language for Web Services
BPML	Business Process Markup Language
BTP	Business Transaction Protocol
CBL	Common Business Language
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
cXML	Commerce Extensible Markup Language
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
ebXML	Electronic Business Extensible Markup Language
EDI	Electronic Data Interchange
FPML	Financial Products Markup Language is a financial XML covering structured financial products.
ICE	Information and Content Exchange Protocol
J2EE	Java 2 Enterprise Edition
Namespaces	
NFS	Network File Services
NIS	Network Information Services
OFX	This stands for Open Financial Exchange. This is the XML specification for exchange of financial data over the Internet.
RPC	Remote Procedure Call
SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
URI	Uniform Resource Identifier is the generic term for all types of names, addresses that refer to objects on the World Wide Web.
WAP	Wireless Application Protocol
WS	Web Services
WSDL	Web Services Description Language
WSEL	Web Services End Point Language
WSFL	Web Services Flow Language
WSIL	Web Services Inspection Language
XLANG	Extensible Language
XML	Extensible Markup Language
XML Base	XML Linking Language needs support linking constructs in a generic way. XML BASE allows authors to explicitly specify a document's base URI for the purpose of resolving relative URIs in links to external images, applets, form-processing programs, style sheets, etc.
XML Information Set	
XML Linking Language	The XML Linking Language defines XML constructs to describe links between resources.
XML Schema	XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the

ACRONYM	Description
	structure, content and semantics of XML documents.
XSL	Extensible Style Sheet Language
XSLT	This stands for Extensible Style Sheet Language Transformations. It is a language for transforming XML documents into other XML documents.

APPENDIX B

WEB SERVICES INDUSTRY CONSORTIUMS

This appendix provides a brief overview of the following three industry consortiums that are involved in the creation of the Web Services standards:

- Organization for the Advancement of Structured information Standards (OASIS)
- World Wide Web Consortium (W3C)
- Web Services Interoperability Organization (WS-I).

The following sections provide a brief history of these three consortiums.

OASIS

OASIS was formed as 'SGML Open' in 1993. At that time, it was focused on developing guidelines for interoperability among products that supported the Standard Generalized Markup Language. In 1998, it changed its name to OASIS to reflect its expanded focus on Extensible Markup Language (XML) and other related standards.

OASIS "provides an open forum where its members discuss market needs and directions, and recommend guidelines for product interoperability. The consortium creates, receives, coordinates, and disseminates information describing methodologies, technologies and implementations of the standards" (OASIS, 2003). It is not a standards body but is focused on "making standards easy to adopt, and the products practical to use in real-world, open system applications. Where appropriate, OASIS recommends specific application strategies over others as ways in which various products can better provide interoperability for users. OASIS helps to apply structured information standards, not create more" (OASIS 2003).

OASIS is open to both corporations and individuals and operates open mailing lists for public comments on specifications. Table 20 summarizes the key events in the history of OASIS.

Table 20: A Brief History of OASIS

Year	Event
1993	"SGML Open" formed to pursue standardization of SGML
1998	"SGML Open" changes its name to OASIS to reflect its new focus
May, 1999	OASIS launches XML.org – the first industry portal to offer an open industry XML registry and repository.
December, 1999	OASIS launches the Electronic Business XML Initiative (ebXML) in partnership with United Nation/CEFACT. OASIS publishes Draft Specification for XML.org Registry & Repository
June, 2000	XML.ORG Goes Live with First Phase of Open Registry & Repository for XML Specifications
February, 2001	OASIS releases the ebXML Technical Architecture Specification ahead of schedule. This specification provides the foundation for all other ebXML specifications.
December, 2001	IBM moves its WSXL standardization work to OASIS. WSUI.org moves its standardization work to OASIS In the security space, AuthXML and S2ML combined their efforts to produce one universally accepted OASIS standard, SAML.
July, 2002	UDDI.org moves its projects to OASIS
November, 2002	OASIS approves the Security Assertion Markup Language (SAML) v1.0 as an OASIS Open Standard.
May, 2003	OASIS approves the Universal Description, Discovery and Integration specification (UDDI) version 2.0 as an OASIS Open Standard.

Source: OASIS (2003)

W3C

W3C was formed in 1994 as a vendor neutral organization and its aim is to develop and promote non-proprietary standards and protocols. Up until the year 2000, it was focused on developing web standards like HTML, PICS and XML.

In April 2001, sensing the need to get involved in the emerging Web Services standards, it conducted a workshop to define the direction for Web Services and to start various work groups to define the various standards that are part of the web services stack. Since then, W3C has been actively driving the creation of Web Services standards. W3C has a well defined process for starting, performing, reviewing and completing projects that lead to standardization of technology. The process is consensus driven and is open for review by the application development community at large. W3C responds to comments from both members and non-members (W3C, 2003). Table 21 summarizes the key events in the history of W3C.

Table 21: A Brief History of W3C

Year	Event
1994	W3C formed to pursue the standardization of Web protocols, particularly http.
1996	W3C approves HTML 3.2 specifications
1997	W3C recommends HTML 4.0 as a standard
1998	W3C recommends XML 1.0 as a standard
1999	W3C issues XSL Transformations (XSLT) and XML Path Language (XPath) as standards
April, 2001	W3C hosts a workshop to plan the future of Web Services officially launching the web services revolution.
July, 2002	W3C publishes First Public Working Draft of Web Services Description Language (WSDL) version 1.2
June, 2003	W3C issues SOAP Version 1.2 as a recommendation.

Source: W3C (2003)

WS-I

WS-I was formed in February, 2002 by Microsoft, IBM, Intel, BEA and HP. The purpose of the group is to "educate businesses on how to build Web services and how to ensure they do it in a compatible way. The consortium will promote existing and future standards defined by the World Wide Web Consortium and the Organization for the Advancement of Structured Information Standards (OASIS)" (WS-I, 2003).

Sun Microsystems was not invited to join the organization and their representative said that they would not join the consortium unless they were given a seat on the Board of Directors. According to IA1MD2, the members of OASIS and W3C worked in the background to convince WS-I to invite Sun to join the group and in October 2002 Sun Microsystems joined the consortium and was given a seat on the Board in March 2003. Table 22 summarizes the key events in the history of WS-I.

Table 22: A Brief History of WS-I

Year	Event
February, 2002	WS-I formed by Microsoft, IBM, BEA, HP and others. Sun was not invited to join the consortium
October, 2002	Sun Microsystems was invited to join the consortium
October, 2002	WS-I released working draft of Basic Profile - a specification that recommends how the core set of web services (SOAP 1.1, WSDL 1.1, UDDI, 2.0 and XML 1.0) should be used to create interoperable web services.
February, 2003	In an industry-wide applauded move, Sun Microsystems decide to delay the release of its J2EE specifications by 3 months to ensure compliance with the interoperability guidelines set forth by WS-I.
March, 2003	Sun Microsystems is elected to the Board of WS-I

Source: WS-I (2003)

APPENDIX C

INTERVIEW GUIDE

The interviews were conducted with three different categories of participants – technology managers in end-user organizations, managers in consulting firms and with industry analysts. A common interview strategy was followed for all the participants. The interview strategy comprised of the following three parts:

- Set the stage with a relevant warm-up opening statement that gives context to the participant.
- Ask a broad all encompassing question that would make them recount all aspects of their experience with middleware.
- Ensure that specific topics of interests are covered while they answer the bigger question.

The following sections provide more details on these three areas for each type of participant and describe the actual script that was used in the interviews.

Technology Managers

Set the Stage (30 seconds or less):

In the last 13 years, we have seen two generations of middleware (DCE and CORBA) come and go. Now we have the 3rd generation, Web Services, which is in the process of being standardized. We also have other technologies that have evolved in this area - .Net is on the way and Java has evolved from a programming language to an infrastructure. You have been in the middle of this all from the very beginning and have done a great job of steering the infrastructure/application development teams through all the gut wrenching changes. In this study, I'm trying to capture your experiences and distill it to come up with something that can be used as a guide to handle similar situations going forward.

Broad Question to elicit maximum coverage of topic:

Can you describe what really happened in your organization with middleware technology? We can start with how and why you selected these (DCE, CORBA and all others you have used), how you went about deploying them, and what really happened when you started using them. We can treat it almost like a post-mortem to get a feel of what we did, why we did it, and what was the result. Also, can you describe your strategy for Web Services?

Specific Areas to be covered while they describe their experience:

Technology Industry Trends and Issues

- What are the key industry trends that have an impact on the organization, operations and IT systems?
- What are the key industry issues that these managers grapple with in their day-to-day work? For example, poor quality of code, uncooperative vendors, software not in compliance with standards, etc.
- What strategies to they adopt to address their issues?
- What is their impression of the role of vendor consortiums?
- What do they believe was done wrong by the vendors and industry consortiums? What was done right?

Internal organizational Factors

- What are the drivers for middleware technology adoption and use?
- Who are the internal stakeholders? What are their issues and priorities?
- What are strategies adopted to deal with the issues?
- Why did a particular technology fail?
- What could they have done differently in each case?
- What is their strategy for Web Services?

Technology Consultants

Set the Stage (30 seconds or less):

In the last 13 years, we have seen two generations of middleware (DCE and CORBA) come and go. Now we have the 3rd generation, Web Services, which is in the process of being standardized. We also have other technologies that have evolved in this area - .Net is on the way and Java has evolved from a programming language to an infrastructure. You have been in the middle of this all from the very beginning and have done a great job of steering your clients through all the gut wrenching changes. In this study, I'm trying to capture your experiences and distill it to come up with something that can be used as a guide to handle similar situations going forward.

Broad Question to elicit maximum coverage of topic:

Can you describe the experience of your clients with middleware technologies, particularly DCE and CORBA? We can start with how and why they selected these (DCE, CORBA and all others available), how they deployed them, and what really happened when they started using them. And also, could you also describe what most of your clients are doing about Web Services today?

Specific Areas to be covered while they describe their experience:

These are same as that for technology managers.

<u>Industry Analysts</u>

Set the Stage (30 seconds or less):
I have been following your research reports for over a decade now. You have done an excellent job of covering the middleware industry since the days of DCE and CORBA. In this study, I'm trying to understand why the middleware technology has been in flux for the last 14 years.

Broad Question to elicit maximum coverage of topic:

What, in your opinion, are the key reasons for the failure of DCE and CORBA? What do you think is the future of Web Services and why?

Specific Areas to be covered while they describe their experience:

- What did OSF and OMG did wrong in their standardization process?
- It is very confusing to see three different vendor consortiums engaged in the development of Web Services standards. They are comprised of the same vendors. What is really happening here?
- Why didn't the key vendors cooperate to make DCE and CORBA successful?
- Why do you think vendors will cooperate this time to make Web Services successful?
- What, in your opinion, makes vendors cooperate to produce a standard?
- What, in your opinion, the entire industry could be doing differently with respect to middleware? I would like to include vendors, end-users, consultants, and industry experts in this question.

APPENDIX D
EXCERPTS FROM AN INTERVIEW TRANSCRIPT

RJ: I'd like to...understand your experiences with DCE and CORBA. Why do you think they failed? What could we have done differently – our organization, vendors and standard bodies? Then [tell me more about] what you are doing with Web Services.

FS1MD2: If we start off with DCE and CORBA, the reason why they didn't succeed was that because the industry was moving too fast. It was putting the horse before the cart. Specifically with respect to really getting people more used to describing the interfaces of the programs they were trying to connect via DCE. While there was IDL, I don't think people had in their DNA to actually develop some sort of contract programming style where you describe your interfaces. Even if you didn't use DCE, but [if] you had a strong process of describing what was expected from a certain interface, results of the inputs and the outputs, [it] would have helped.

I also think we were pushing hard on the infrastructure that was promised all things to all people but the people hadn't gone through the cultural change of understanding of what it means to really develop contract based programming styles where the interfaces are well understood and the expectations of the interfaces are well understood.

And [another] reason was that DCE was sort of monolithic. It was not as modular as it ought to have been... You needed the security and the directory services...to be there for it to work. It was intended to be a comprehensive solution but since you could not deploy [these] capabilities in a modular fashion, [it] made it somewhat impossible to use.

I think the industry went wrong in that the vendors, such as Sybase, did not get behind it all the way and did not provide enough support for the technology. So, while the solution was promised, the fact was that many of the vendors on whom we relied for core technology were not really in the game.

RJ: So this was the problem of that I have a crucial product not integrating with the core infrastructure.

FS1MD2: And because [these vendors] were not willing to integrate with DCE.

I think the OSF did not sell well. They thought that [if they] build [the products, other vendors] would come [and support the new technology]. They did not persuade the vendors or explain to them the benefit of joining the bandwagon.

Many of these vendors were very well established in their own right and they had alternate ways of connecting [the applications together]. They had alternative models of distribution [and] of integration. It was not clear why should they buy into [DCE]. So the ground work to make sure that the industry was getting broader support especially from other crucial products and vendors [was not done].

RJ: You mentioned the paradigm shift for the developers. They were doing client-server before too.

FS1MD2: Yes. In my view, this is critical for web services too.

The notion of people describing interfaces to their own collateral of code is very important...Describing the interfaces...the expected input and output, and the conditions [under which the interfaces are called]...and going through that rigor hasn't been really the domain of AD folks. AD folks are [used to] describing an RPC.

DCE ...focused a lot on making sure that people describe the interfaces to the services that they provided... And AD developers are still not doing that well.

That's what I call contract based programming. Essentially people say this is what I promised to deliver. Then once you make these [function] calls, once you make these requests, these are the inputs, these are the outputs, these are the exceptions, [and] these are the issues.

CORBA also presented the same kind of interface but people again are not very disciplined about how they describe their interfaces. And that to me is the heart of the service oriented architecture...you are describing what services you are going to offer.

RJ: So in this case we did have an architecture.

FS1MD2: We were the architecture team and we then worked with the infrastructure team to figure out how the infrastructure would be deployed.

The developers did not...adopt...The fact [was] that everybody was able to do easily with Sybase [what we were asking them to do with DCE or CORBA]. So why on earth would [an application developer] go to some other environment [when all their code was] already [working] on Sybase. And if [they] needed a multi-tier solution, [they could have used] the Sybase open server. So there were alternative ways of getting there.

RJ: So you had both the architecture and infrastructure in place but the AD buy-in did not happen?

FS1MD2: Yes

RJ: But we had training classes in place to train the developers. So what really happened?

FS1MD2: So that was my point about putting the cart before the horse. OSF worked hard on the technology but did not work hard on the paradigm shift of the coding or programming rules. It didn't work hard on the people. What people saw was that DCE was about RPC and they thought that they could use Sun RPC and use Kerberos for security. So why should [they] use DCE?

So going back to buy-in, there were two levels of buy-in. One, there was buy-in around are these products real.

RJ: was it about the quality of the product?

Tays: No, it was more...about the key vendors I depend upon. Will DCE integrate with their products? And the other level, the higher level, was that people mentally were not used to describing what they were doing according to well defined interfaces. People were not doing that. It was a big paradigm shift. And my view is that if the industry doesn't crack that even Web Services will not be successful.

RJ: Now one thing that has changed [over time is the perception of business people about middleware]. So if Web Services is the new reincarnation of middleware, then I'm curious to find out whether you think that you have more buy-in from the business now for Web Services than you had for DCE and CORBA, which they had never heard of?

FS1MD2: No. Frankly, the businesses don't give a toss. My view is that the businesses are saying that this is the method techhies are talking about to do integration. When they think middleware, they think that [it is] going to help [them] link applications and databases together. Generically, that's [all] they can see. . I will be surprised if anybody says that a business person really does care about it. I'd frankly like to challenge them. This is really the domain of the IT folks. And this is coming from someone who believes very passionately about getting the opinion of the businesses.

Web services...we know it will help...but all this doesn't have any more significance than that. We know Microsoft is talking about it. [Other big vendors] are talking about it. But it hasn't translated yet [into something concrete]. Think about the Internet as a paradigm. Also think about the Internet as a distributed middleware environment. The thing about the Internet is that people can relate to it because ultimately it enables them to access information very simply and easily. It is simple and same everywhere. And it feels like that too. And honestly, any middleware environment that doesn't have these features - simplicity, ease of access to information, wherever that information is, will not work. Because in the mind of a business person, middleware is like the Internet.

So, Web Services, to the businessperson is a more improved environment for connecting to the world.

RB: They still do not think of it as something that may change things for them or give them a strategic advantage for the firm.

FS1MD2: Maybe I am missing something here, but even I'm not sure if Web Services gives me a strategic advantage. Because I have gone through the experience of DCE and CORBA...I'm thinking what's different this time other than there is perhaps more integration with the Internet standards like XML, SOAP, and UDDI...So I'll really treat it as something that will make it simpler for programmers to connect to services that may be distributed in Internet architecture or non-Internet architecture. The struggle with Web Services is that you have to make sure that people understand that it is not a panacea. This is not about RPCs. This is about describing interfaces to services that you are making available so that other people can tap into them.

RJ: So this is the next level up from DCE.

FS1MD2: Web services have much more richness in describing the interfaces. It also has richness with respect to a lot of the underlying standards it depends on, i.e. the Internet standards.

And therefore, unlike previously, when OSF... and OMG were just pushing [their technology and expecting] everybody to join them whether they like it or not; this time it has taken a longer process but there has been much more buy-in from the major vendors around all these technical pieces...[This is] because we have leveraged the Internet standards process. That's why the Web Services has a much better chance of success.

However, it does put...on the programmers a strong sense of responsibility of describing the interfaces properly so that the code [they] develop can be used by others. All that would translate in the end [into increased] productivity.

So I think that we should be careful about overstating and over hyping about what it means in terms of strategic advantage because that has hurt us in the past. Because I think personally that the real benefits come from easier access to information and services that are just rightly distributed. As long as I can do that, then I can get some advantage

RJ: So going back to DCE as the first generation and CORBA as the second generation and knowing what you know now, how would you have done things differently?

FS1MD2: I think what I'd do differently goes to some of the things I had mentioned earlier. We should not underestimate the power of the vendors who produce critical pieces of technology you depend on, particularly database and desktop vendors. And if they are not in the game, [then] you can have [standard] bodies, competitors like Lehman and Citibank [show a lot of interest]...but, if the key [vendors] are not in the game [then] we are not going to be successful because our legacy [applications are] dependent on their products.

Even IBM, in all its might, was involved in DCE. Unfortunately they did not integrate [DCE] with their own database platform...But if Oracle and Sybase had done it and if other critical vendors had done it, then it could have had a chance. And that's the difference now. Sybase and Oracle are doing Web Services. They believe in this stuff.

RJ: Do you think you have a more of a developer buy-in this time with web services than with DCE and CORBA.

FS1MD2: Yes, I'd say that. And I think this is really connected to the success of the Internet. Putting the so called [internet] bubble aside, [it is] related to the ability to get on the web and get access to the information [you need]...People believe in it. The real concern I have around Web Services, [and] it is really a question of time, is that there are still a lot of pieces of the puzzle that still have to fall in place. A robust UDDI [and] security is still not there. And because we have seen so much innovation in the IT industry, and the world is not going to stand still, my concern is that if we are not going to crack all the integration and security pieces and really make is smooth like DCE [it will fail]. DCE was smooth but difficult to fit into place and configure.

Can we really move quickly to put in place the infrastructure for Web Services to be available? Otherwise what may happen is that some other guy may start yet another distributed paradigm! Someone may say that "Oh! They haven't solved these problems, and we will go in and solve it." Then you have a split in the industry.

So while I agree that we have done a better job of buy-in at the grass roots level and across vendor organizations...the real concern is [whether] the industry is moving fast enough.

RJ: So what are you doing across your organization to handle Web Services? What is your strategy?

FS1MD2: We are doing a number of things. We are trying to be very pragmatic about it...We have a very strong legacy environment. We have a number of systems that have been around for years. They run well, scale well, and so we have a concern about saying that everything will be Web Services. But what we do want to do is to make sure that people can wrap [the legacy systems as Web Services]. So one of the first things we did was [that] we embarked on an XML standardization effort. [This will] make sure that...any channel of interaction [we have] with our customers [will] be able to link up with any application internally.

RJ: So you took the first step of standardizing the data interchange protocol.

FS1MD2: Exactly. So we standardized the information delivery architecture by saying that we have to have standardized messages with standard SOAP headers. That was the first thing. Second thing we will do, as we begin to introduce reusable components, we [will] make sure that the interfaces are well defined.

RJ: When you say "reusable components", are you redesigning the applications?

FS1MD2: In some cases, some of the applications are being redesigned and being broken into smaller components. In those cases, we are making sure that they use XML and that they describe these new services and components via web services interfaces

RJ: What technologies are you using for these?

FS1MD2: We are using J2EE, java platform, BEA web logic, etc.

RJ: We were talking about the strategy you had adopted - you standardized on the data exchange protocols.

FS1MD2: We also wanted to make sure that we were leveraging more of the J2EE platform to actually help us build distributed applications and also help us get to web services frame work.

RJ: So you looked at the framework, [analyzed] what things you can do now without impacting anything, and that will give you an immediate gain. And then you can wait and see as things evolve and then tie them together.

FS1MD2: Yes

RJ: That's what we should have done with DCE.

FS1MD2: As I said, DCE was very monolithic and not modular. I really feel that in web services, we have learnt a lot from the Internet world and that there is a good buy-in from the people.

RJ: What are some of the key challenges that you have in your role?

FS1MD2: My key challenge right now is to make it simpler for people to get to a holistic view of information about customers [and] organization. How we get applications to link up together and to [be] able to exchange information without necessarily consolidating all these platforms into one. That is the primary challenge. So interoperability, exchange of information, and security of information are the vital things we are really trying to focus on.

RJ: So how are you organized to handle this?

FS1MD2: We have lines of business and we have IT groups aligned to them. However, each line of business has a responsibility for the infrastructure we have, although we get those services from the centralized organization. We also have cross business groups like architecture, warehousing, etc. to take advantage of cross line of business (LOB), kind of things. So we provide those functions across businesses. And we also lead the information strategy around XML and so on. So we have the LOB IT teams that are delivering IT solutions to the LOBS and we have cross-business groups to make sure that we are working on common issues.

APPENDIX E

ATLAS.TI SOFTWARE DETAILS

ATLAS.ti is a software package provided by Scolari, the software division of Sage Publications. The details in this appendix are a copy the marketing material provided on the company web site (http://www.scolari.co.uk).

It is a powerful software package for the visual qualitative analysis of large bodies of textual, graphical and audio or video data. It offers a variety of tools for accomplishing the tasks associated with any systematic approach to "soft" data. ATLAS.ti helps uncover the complex phenomena hidden in qualitative data, offers a powerful and intuitive environment for coping with the inherent complexity of tasks and data, and keeps the researcher focused on the data under analysis. It provides the following functionality:

- collect and organize text, audio and visual data files, along with the researchers coding, memos and findings, into a project or "hermeneutic unit",
- facilitate the activities involved in analysis and interpretation, in particular selecting, coding, annotating, and comparing relevant data segments,
- provide a comprehensive overview of the research; as well as rapid search, retrieval and browsing of all data segments and notes relevant to an idea,
- build unique networks which allow the researcher to "connect" visually selected passages, memos, and codes. This enables the researcher to construct concepts and theories based on visible relations and to reveal other relations.

ATLAS.ti is being used extensively in areas characterized by a systematic, yet creative, approach to analyzing both unstructured and semi-structured data. The following are some of the fields in which researchers are using ATLAS.ti:

- Software Engineering
- Management studies
- Economics
- Anthropology
- Psychology
- Law
- History
- Linguistics
- Criminology

ATLAS.ti is available in PC version only, running on Windows 95 or later, or Windows NT 4.0. ATLAS.ti also runs on the following configurations:

- Apple Macintosh™ with VIRTUAL PC installed.
- PCI based Sun Ultra Workstations with SunPCI Copressor

The recommended hardware configuration for ATLAS.ti is 500 MHz processor, 64 MB or more of RAM, 45 MB or more hard disk, and sound card for processing audio data.

APPENDIX F

CODES, CATEGORIES & THEIR RELATIONSHIPS

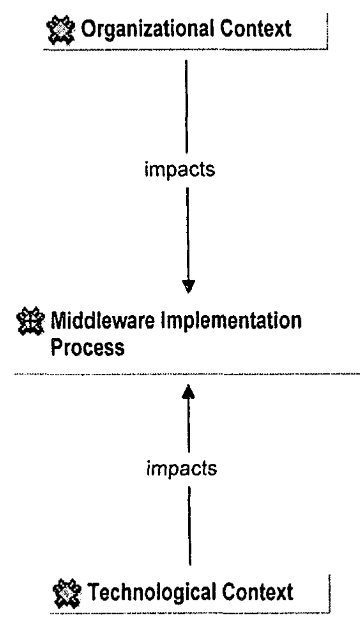

Figure 27: Middleware Implementation Framework Core Categories

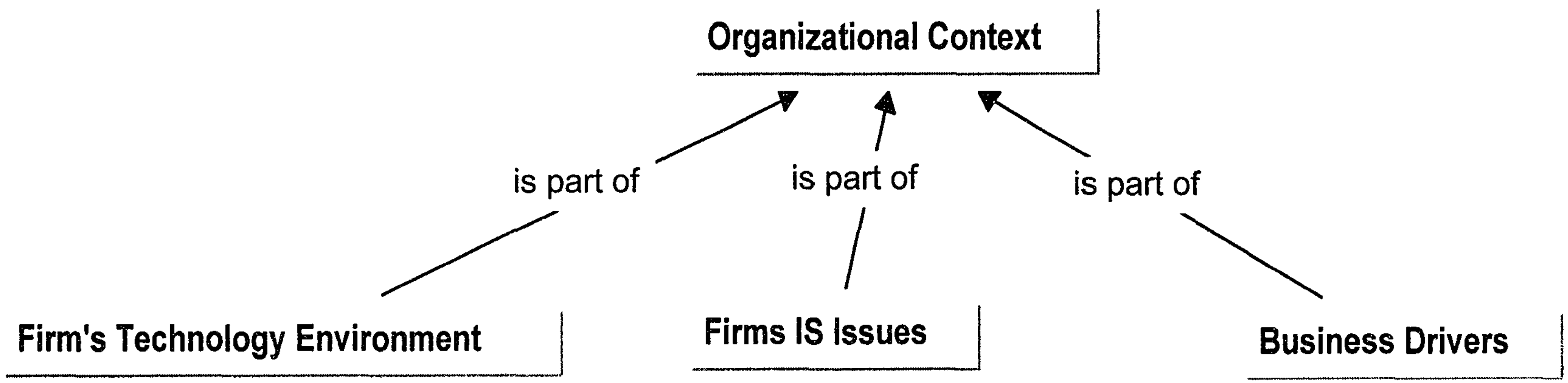

Figure 28: Sub-categories for Organizational Context

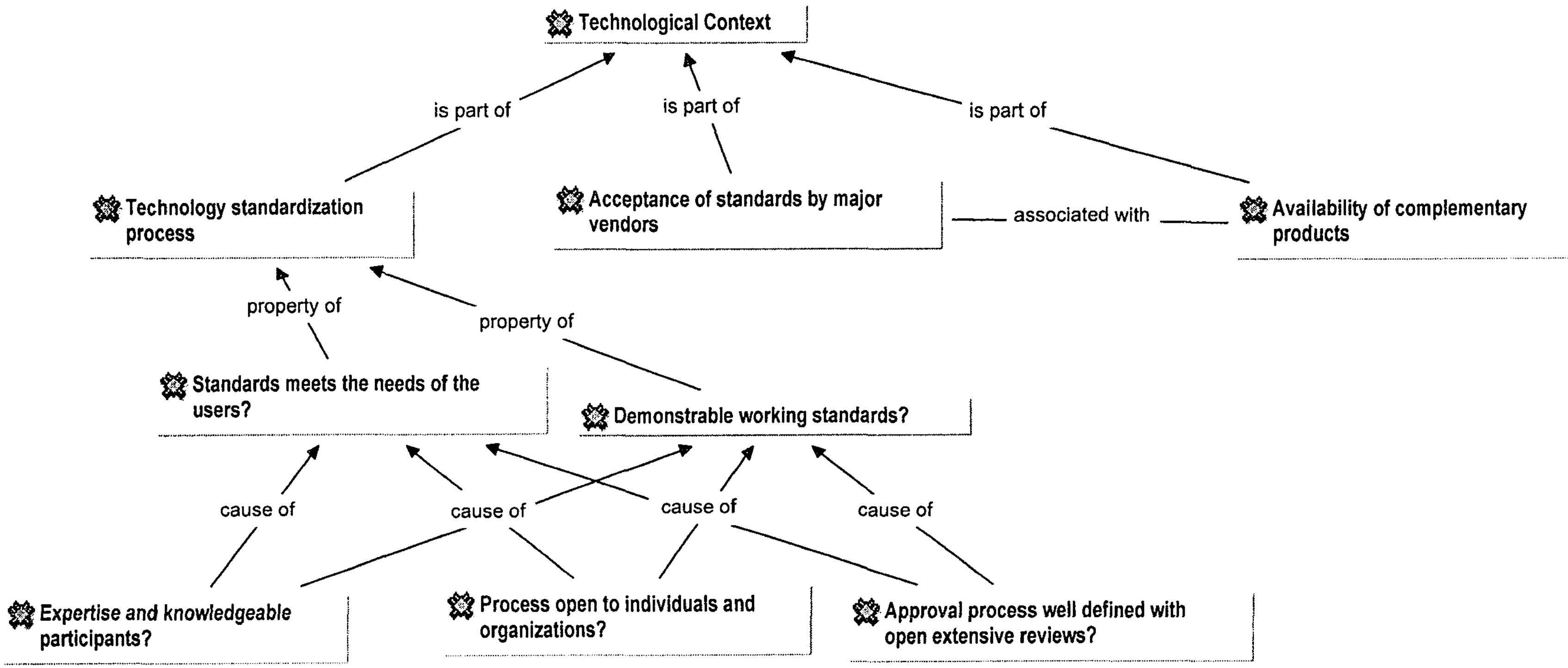

Figure 29: Sub-categories for Technological Context

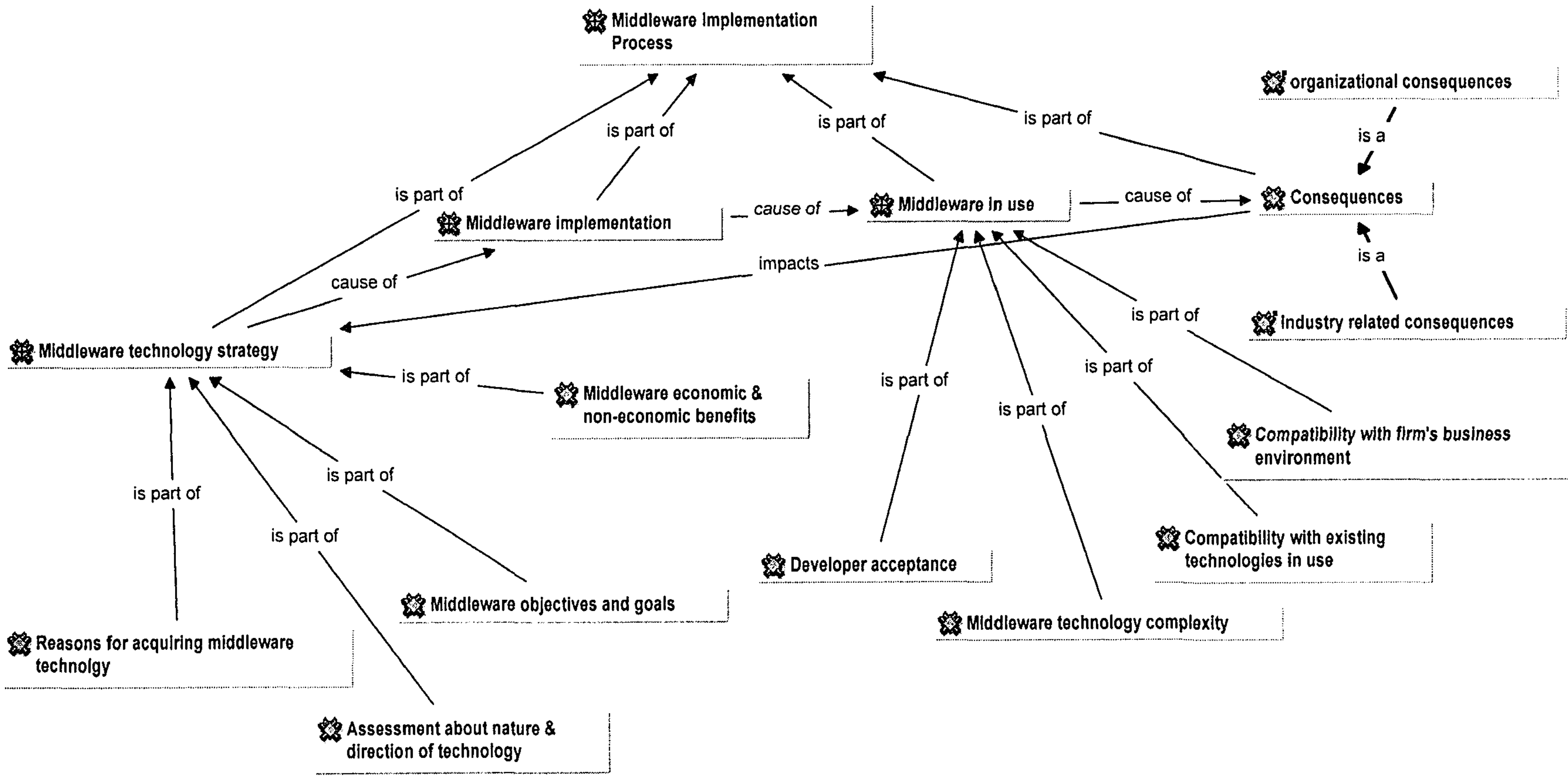

Figure 30: Sub-categories for Middleware Implementation Process

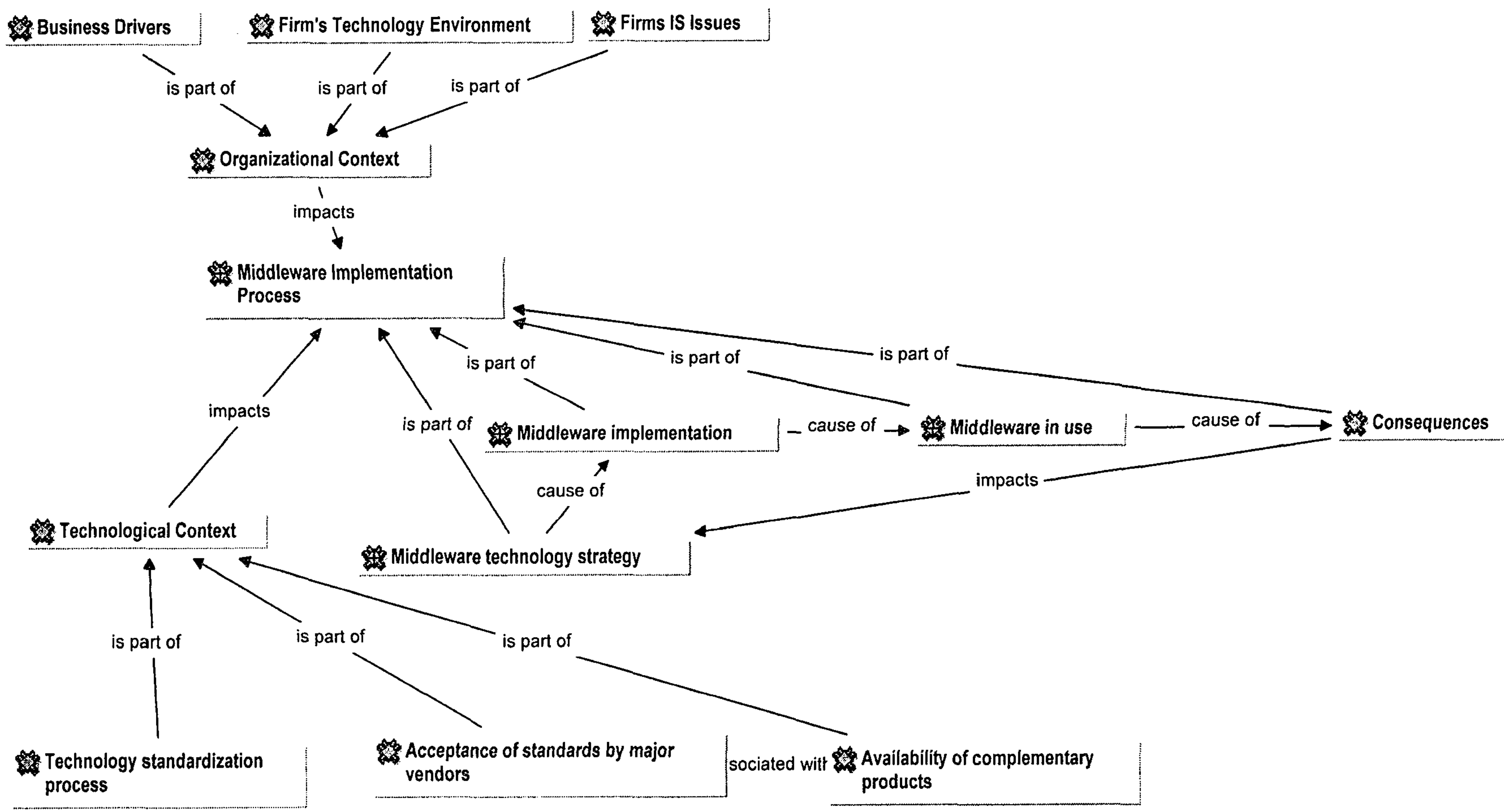

Figure 31: Sub-categories for Middleware Implementation Framework

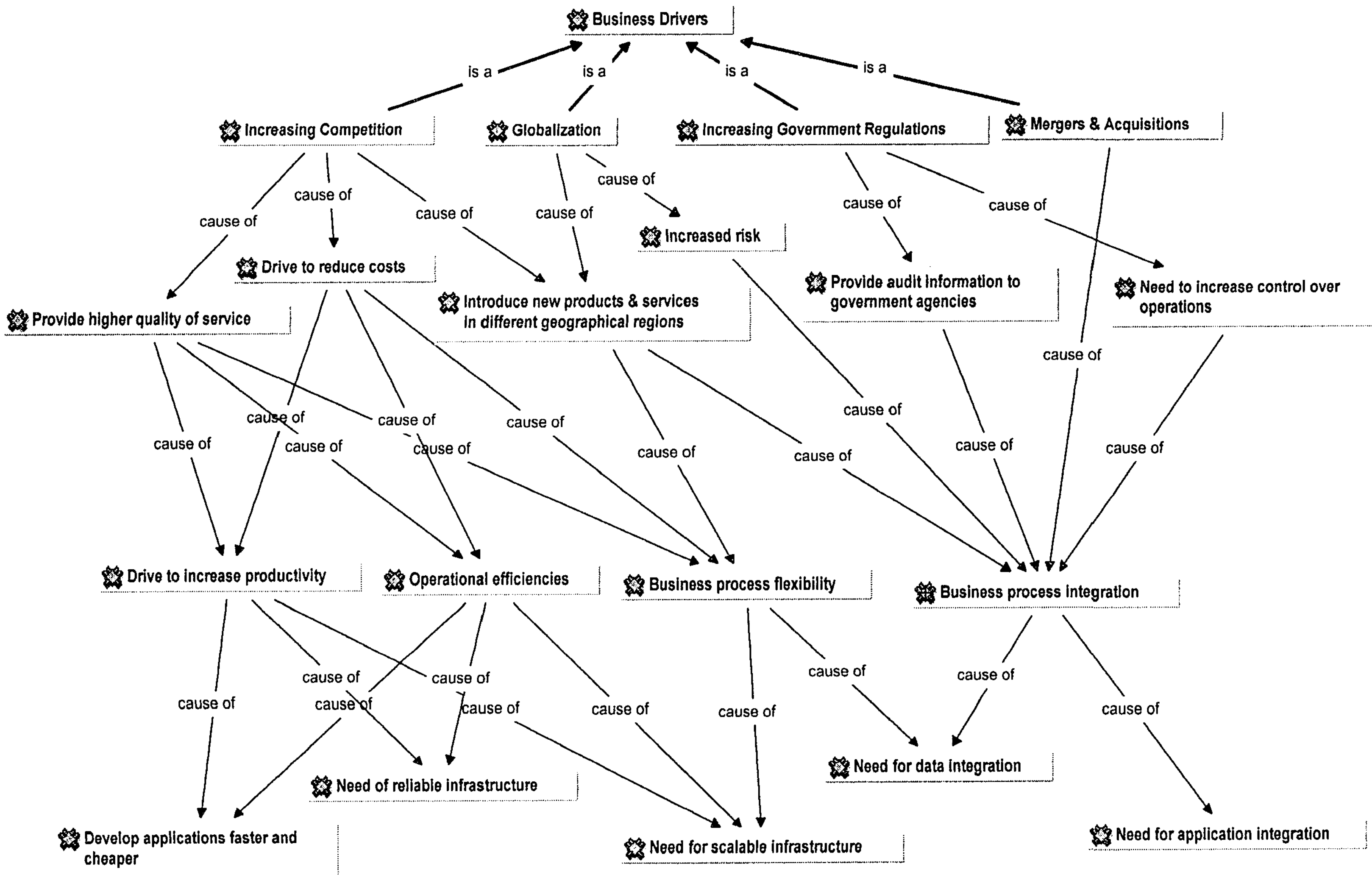

Figure 32: Sub-categories for Business Drivers

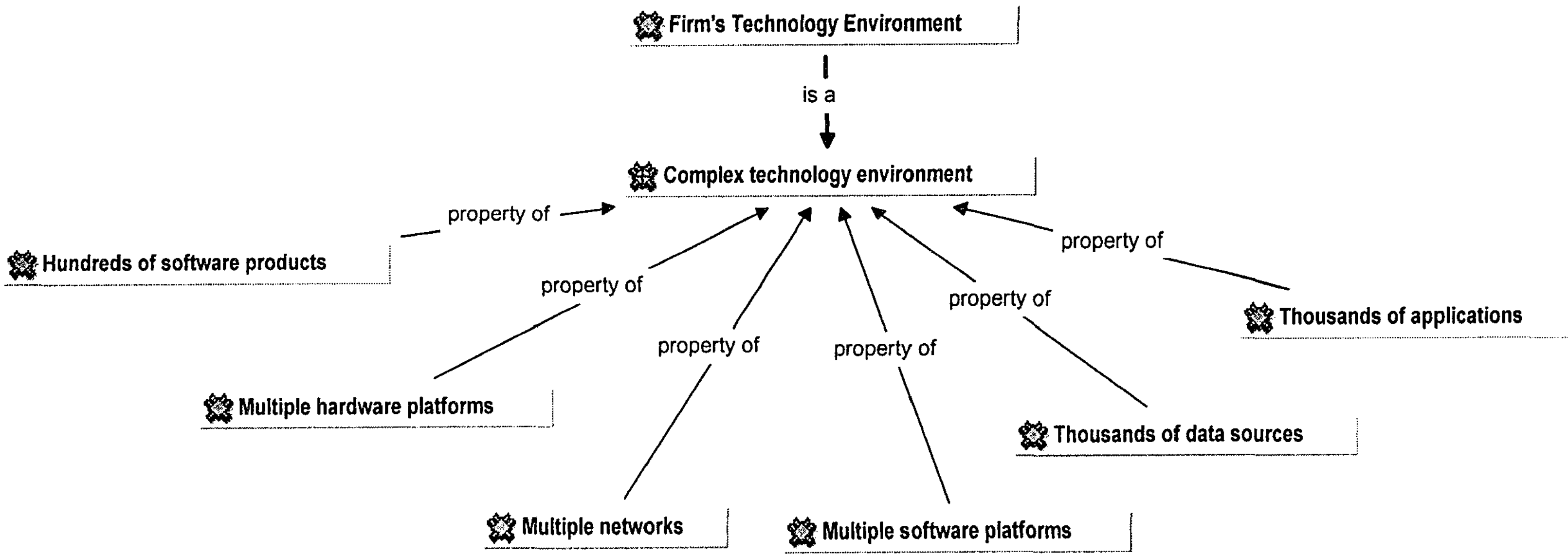

Figure 33: Sub-categories for Firm's Technology Environment

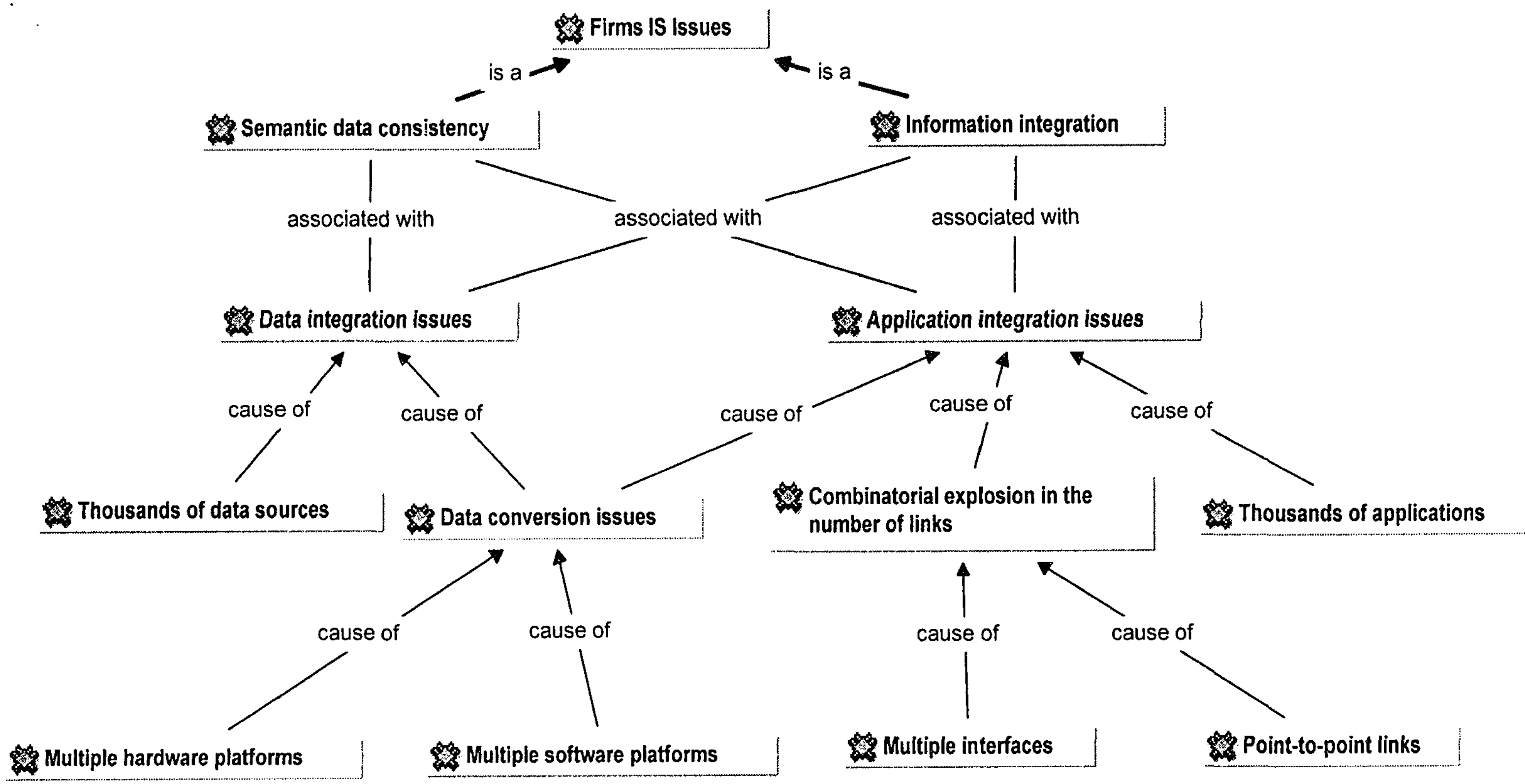

Figure 34: Sub-categories for Firm's IS Issues

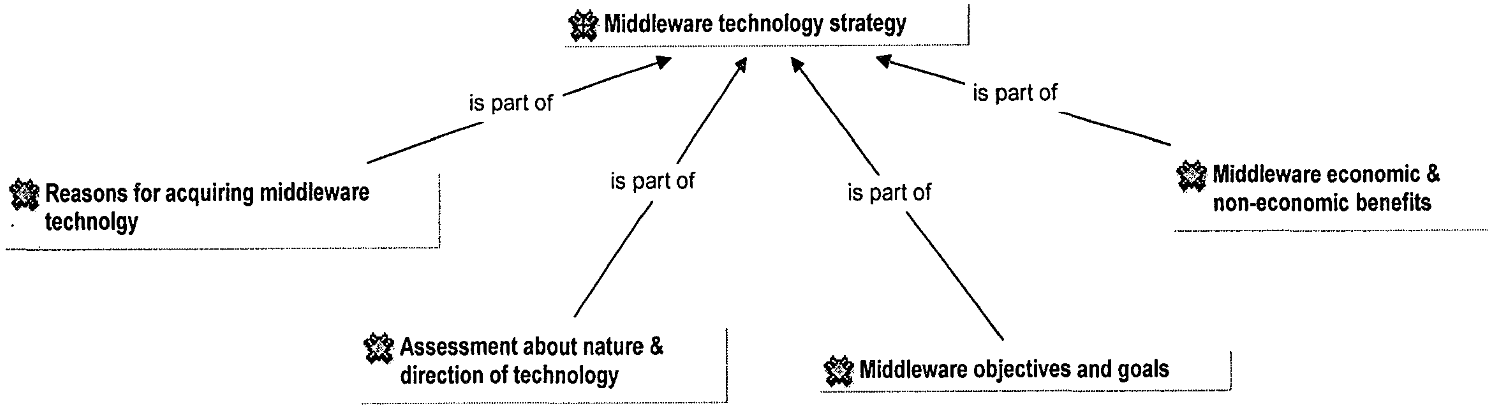

Figure 35: Sub-categores for Middleware Technolgy Strategy

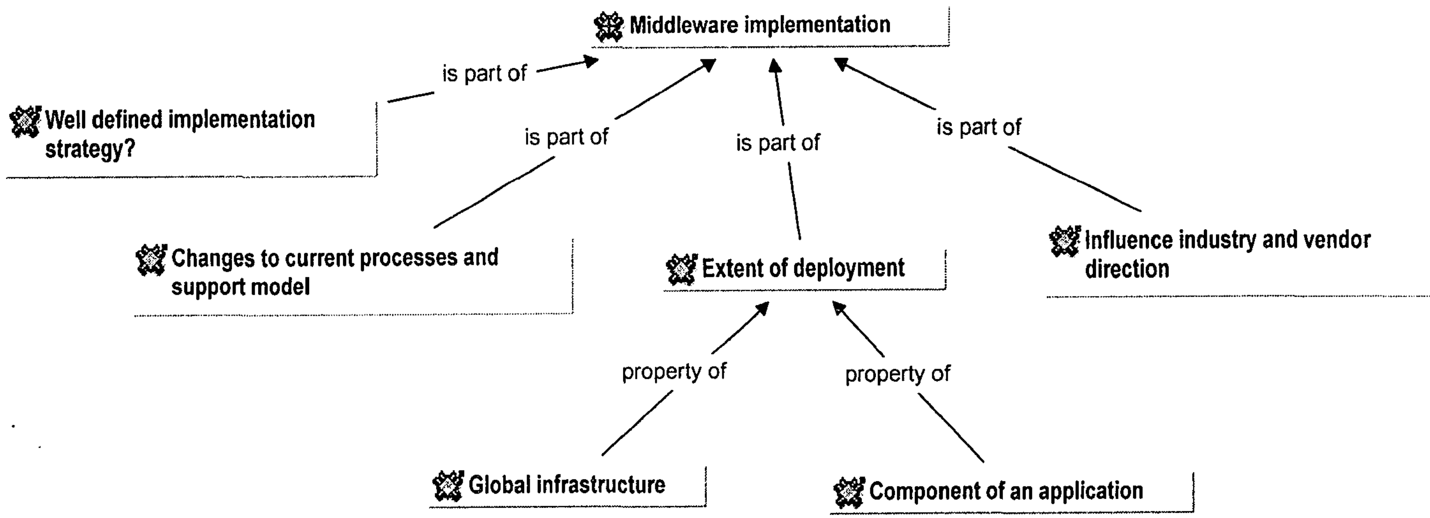

Figure 36: Sub-cagtegories for Middleware Implementation

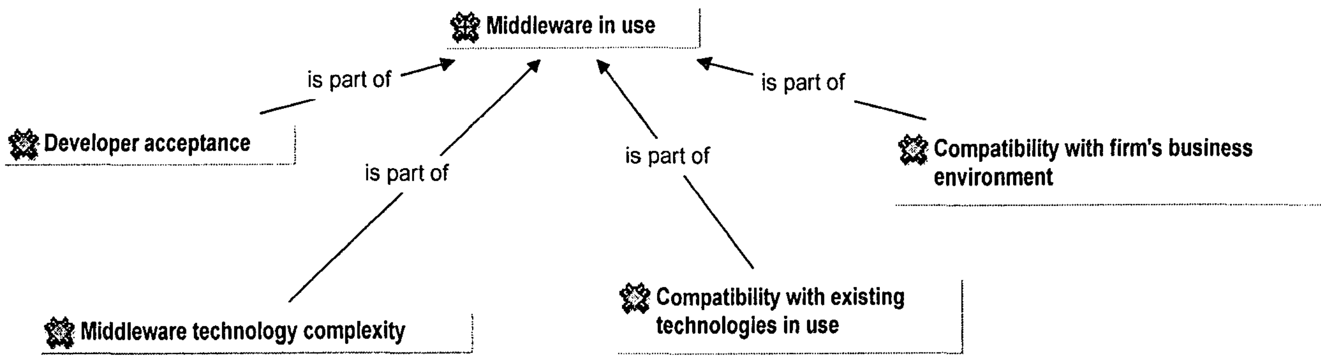

Figure 37: Sub-categories for Middleware in use

REFERENCES

Abernathy, W.J., & Utterback, J.M. "Patterns of Industrial Innovation." *Technology Review*, June-July, 1978, pp. 40-47.

Adam, L. & Wood, F. "An Investigation of the Impact of Information and Communication Technologies in Sub-Saharan Africa." *Journal of Information Science,* 25(4), 1999, pp. 307-318.

Alavi, M., & Joachimsthaler, E. A. "Revisiting DSS Implementation Research: A Meta-Analysis of the Literature and Suggestion for Researchers." *MIS Quarterly*, 16(1), 1992, pp. 95-116.

Allen, Paul, Editor. *Component Development Strategies*, by: Cutter Information Corp, VOL. XII, NO. 1. January, 2002.

Anderson, P., & Tushman, M.L. "Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change." *Administrative Science Quarterly,* 35, 1990, pp. 604-633

Ang, J.S., Sum, C.C., & Chung, W.K. "Critical Success Factors in Implementing MRP and Government Assistance, a Singapore Context," *Information and Management*, 29, 1995, pp. 63-70.

Bensen, S.M., & Farrell, J. "Choosing How to Compete: Strategies and Tactics in Standardization." *Journal of Economic Perspectives*, 8(2), 1994, pp. 117-132.

Brancheau, J. C., Janz, B.D., and Wetherbe, J. C. "Key issues in information systems:1994-95 SIM Delphi results," *MIS Quarterly*, 20, 2, 1996, pp. 225-243.

Cale, E.G., & Eriksen, S. E. "Factors Affecting the Implementation Outcome of a Mainframe Software Package: A Longitudinal Study," *Information and Management,* 26(3), 1994, pp. 165-175.

Calloway, L.J. and Ariav, G. "Developing Using a Qualitative Methodology to Study Relationships Among Designers and Tools," in Nissen, H.E., Klein, H.K. & Hischheim, R. (Eds.), *Information Systems Research: Contemporary Approaches and Emergent Traditions*, Amsterdam, Holland, 1991

Chau, P.Y.K. "An Empirical Investigation on Factors Affecting the Acceptance of CASE by Systems Developers," *Information & Management,* 30(6), 1996, pp. 269-280.

Chau, P.Y.K., & Tam, K. Y. "Factors Affecting the Adoption of Open Systems: An Exploratory Study," *MIS Quarterly*, 1997.

Chau, P.Y.K., and Tam, K.Y. "Organizational Adoption of Open Systems: A 'Technology-push, Need-pull' Perspective," *Information and Management*, 37, 2000, pp. 229-239.

Chengalur-Smith, I., and Duchessi, P. "The Initiation and Adoption of Client-Server Technology in Organizations ", *Information & Management*, 35(3), 1999, pp. 77-88.

Christensen, C.M., Suarez, F.F., & Utterback, J.M. "Strategies for survival in fast changing industries." *Management Science*, 44(12), 1998, pp. 207-220

Cooper, R., Zmud, R.W. "Implementation Technology Information Research: A Technological Diffusion Approach," *Management Science*, 36(2), 1990, pp. 123-139.

Correia, Joanne M., Biscotti, F. & Contu, R. *AIM and Portal Software Share and Forecast: Vertical Industries, 2002*. SOFT-WW-EX-0140. Gartner. 17 November, 2003.

Creswel, J.W. *Qualitative Inquiry and Research Design*, Thousand Oaks, CA: Sage Publications, 1998.

Curley, K.F, & Gremillion, L.L. "The Role of the Champion in DSS Implementation," *Information and Management*, 6(4), 1983, pp.203-209.

Davis, F. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology", *MIS Quarterly*, 13(3), 1989, pp. 319-340.

Davis, F., Bagozzi, R., and Warshaw, P. "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models", *Management Science*, 35(8), 1989, pp. 982-1003.

Dessauer, J. H. *My Years with Xerox*, Manor Books, New York: 1975.

Earl, M.J., "Experiences in Strategic Information Systems Planning," *MIS Quarterly*, 17:1, March 1993, pp. 1-24.

Eisenhardt, K.M., "Building Theories from Case Research," Academy of Management Review, (14:4), October 1989, pp. 532-550.

Eisner, E. *The enlightened eye: Qualitative inquiry and the enhancement of educational practice*, Macmillan, New York: 1991.

Essex, P.A., Simha, R.M., and Masteller, D.E. " Determinants of Information Center Success," Journal *of Management Information Systems*, 15(2), 1998, pp. 95-117.

Farrell, J., & Saloner, G. "Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation." *American Economic Review*, 76(5), 1986, pp. 940-955.

Farrell, J., & Saloner, G. "Standardization, Compatibility and Innovation." *Rand Journal of Economics*, 16, 1985, pp. 71-83.

Galal, G.H. "From Contexts to Constructs: The Use of Grounded Theory in Operationalising Contingent Process Models," *European Journal of Information Systems*, 10(1), 2001, pp. 2-14.

Gilpin M., Narsu, U. and Randy Heffner, R. *Web Services: Emerging as a Broad Internet Application Integration Standard, Bigger Than Just Microsoft .NET.* : June 2001. Giga Information Group.

Gilpin, Mike. *Web Services Hype Backlash: Balance Skepticism with Cautious Optimism.* Giga Information Research. July 2002

Glaser, B.G., & Strauss, A.L. *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine Publishing Company, New York, 1967.

Gordon, S., & Gordon, J. "Organizational Hurdles to Distributed Database Management Systems (DDBMS) Adoption," *Information and Management*, 22,1992, pp.333-345.

Grover, V., & Teng, J. "An Examination of DBMS Adoption and Success in American Organizations," *Information and Management*, 23, 1992, pp.239-248.

Guimaraes, T., Igbaria, M., & Lu, M. "The Determinants of DSS Success: An Integrated Model," *Decision Sciences*, 23(2), 1992, pp.409-430.

Guimaraes, T., Yoon, Y. and Clevenson, A. "Factors Important to Expert System Success - A Field Test," *Information and Management*, 30, 1996, pp.119-130.

Hayes-Roth, F. and Jacobstein, N. "The State of Knowledge-Based-Systems", *Communications of the ACM*, 37(3), 1994,pp. 27-39.

Henderson, J. C. and Schilling, D. A. "Design and Implementation of Decision Support Systems in the Public Sector," *MIS Quarterly*, 9(2), 1985, pp.157-170.

Hoffer, J.A., & Alexander, M.B. " The diffusion of database machines", *Data Base*, 23(2), 1992, pp. 13-19.

Hutchinson, S. A. "Education and Grounded Theory" In R. Sherman and R. B. Webb, (Eds.), *Qualitative Research in Education: Focus and Methods*, The Falmer Press, New York, NY: 1988.

Katz, M. L., & Shapiro, C. "Network Externalities, Competition, and Compatibility." *American Economic Review*, 75(3), 1985, pp. 424-440.

Katz, M. L., & Shapiro, C. "Product Introduction with Network Externalities." *Journal of Industrial Economics*, 40(1),1992, pp. 55-83.

Katz, M. L., & Shapiro, C. "Systems Competition and Network Effects." *Journal of Economic Perspectives*, 8(2), 1994, pp. 93-116.

Katz, M. L., & Shapiro, C. "Technology Adoption in the Presence of Network Externalities." Journal *of Political Economy*, 94(4), 1986, pp. 822-841.

Kunnathur, A.S., Ahmed, M.U., and Charles, R.J.S. "Expert Systems Adoption: An Analytical Study of Managerial Issues and Concerns," *Information and Management*, 30, 1996, pp. 15-25.

Kwon, T., and Zmud, R.W. Unifying the fragmented models of information systems implementation in R. Boland, R. Hirscheim (Eds.), *Critical Issues in Information Systems Research*, John Wiley & Sons, New York, 1987, pp. 227-251.

Lai, V.S. "Critical Factors of ISDN Implementation: An Exploratory Study", *Information and Management*, 33(2), 1997, pp. 87-97.

Lai, V.S., & Guynes, J.L. "A Model of ISDN Adoption in U.S. Corporations", *Information and Management*, 26, 1994, pp. 75-84.

Lai, V.S., and Reeh, B. "ISDN implementation in the United States and Germany: A Cross-Country Assessment," *Information and Management*, 29, 1995, pp.141-140.

Lanowitz, Theresa. *Quality Development Is Crucial to Emerging Technologies*. Gartner. April, 2003.

Lee, D.S. "Usage Patterns and Sources of Assistance to Personal Computer Users," *MIS Quarterly*,10(4), 1986, pp. 313-325.

Lee, J.R., O'Neal, D. E., Pruett, M. W., & Thomas, H. "Planning for dominance: a strategic perspective on the emergence of a dominant design." *R&D Management*, 25(1), 1995, 3-25.

Lieberman, M., and Montgomery, D. "First mover advantages: A survey", *Strategic Management Journal*, (9), 1988, pp. 41-58.

Lincoln, Y., and Guba, E. *Naturalistic Inquiry*, Sage Publications, Beverly Hills, CA: 1985.

Lucas, H.C., and Spitler, Valerie. "Implementation in a world of workstations and networks," *Information and Management*, 38, 2000, pp. 119-128.

Magal, S. R., Carr, H. H., and Watson, H. J. "Critical Success Factors for Information Center Managers", *MIS Quarterly*, 12(3), 1988, pp. 413-425.

Mansfield, E., Rapoport, J., Romeo, A., Villani, E., Wagner, S., and Husic, F. *The Production and Application of New Industrial Technology*, Norton, New York: 1977.

May, K.A. Writing and evaluating the grounded theory research report. In W.C. Chenitz & J.M. Swanson (Eds.), *From practice to grounded theory* (pp. 146-154). Edison-Wesley, Melno Park, CA: 1986.

Meyer, M.H., and Curley, K.F. "Putting Expert Systems Technology to Work," *Sloan Management Review*, Winter 1991, pp. 21-32.

Moore, Stephanie. *OMG's CORBA: 1996 Is the Pivotal Year*. Giga Information Group. February, 1996.

Moores, T.H. "Key Issues in the Management of Information Systems: A Hong Kong Perspective", *Information and Management*, 30(6), 1996, pp. 301-307.

Morrow, S.L. and Smith, M.L. "Constructions of survival and coping by women who have survived childhood sexual abuse," *Journal of Counseling Psychology*, 42, 1995, pp. 24-33.

Narsu, U, Vice President. *XML Standards and the future of Web Services*. Giga World IT Forum, 2002. PowerPoint Slides.

Natis, Yefim and Jess Thompson. *What Can Web Services Do for You*. Gartner, July, 2001.

Natis, Yefim. *Application Server Scenario: From Stovepipes to Services*. Gartner. October, 2001.

Niederman, F., Brancheau, J.C., and Wetherbe, J.C. "Information Systems Management Issues for the 1990s," *MIS Quarterly*, 15(4), December 1991, pp. 475-500.

OASIS. [Online] Organization for the Advancement of Structured information Standards, Cited September 25, 2003. Available from <http://www.oasis-open.org>.

Ogbuji, U.[Online] The Past, Present and Future of Web Services, Cited August 11,2003, Available from <http://www.webservices.org/index.php/article/articleview/663>, September, 2002.

OMG. [Online] The Object Management Group, Cited September 25, 2003. Available from http://www.omg.org, 2003.

Open Software, Open Software Foundation. *Introduction to OSF DCE 1.1*. Pearson Education POD, UK: 1995.

Orlikowski, W. "CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development," *MIS Quarterly*, 17(3), 1993, pp. 309-340.

Orlikowski, W. and Gash, D. "Technological Frames: Making Sense of Information Technology in Organizations," *ACM Transactions on Information Systems*, 12(2), April, 1994, pp. 174-207.

OSF. [Online] The Open Group, Cited September 23, 2003. Available from http://www.opengroup.org, 2003

Palvia, S.C., and Chervany, N.L. "An Experimental Investigation of Factors Influencing Predicted Success in DSS Implementation," *Information and Management*, 29, 1995, pp.43-53.

Preece, D.A. "They Whys and Wherefores of new technology adoption," *Management Decision*, 29(1), 1991, pp. 53-58.

Ramamurthy, K. & Premkumar, G. "Determinants and outcomes of electronic data interchange diffusion," *IEEE Transactions on Engineering Management*, 42(4), 1995, pp. 332-351.

Robey, D., Ross, J.W., and Boudreau, M.C. "Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change," *Journal of Management Information Systems* (19:1), Summer 2002, pp. 17-46.

Rogers, E.M., and Shoemaker, F.F. Communication of Innovations: A Cross-Cultural Approach, The Free Press, New York: 1971.

Rogers, E.M., The Diffusion of Innovations, The Free Press, New York: 1983.

Rudelius, W., Dickson, G. W., & Hartley, S. W. "The Little Model that Couldn't: How a Decision Support System for Retail Buyers Found Limbo," Systems, Objectives and Solutions, 2(3), 1982, pp. 115-124.

Rutkowski, A. M. "Today's Cooperative Competitive Standards Environment and the Internet Standards-Making Model." In B. Kahin & J. Abbate, *Standards Policy for Information Infrastructure,* 1995, Cambridge, MA: MIT Press.

Rymer, John. *Lacking Support from ISVs, DCE will Fade in Enterprise Integration.* The Meta Group, 1997.

Sanders, G.L., and Courtney, J. F. "A Field Study of Organizational Factors Influencing DSS Success," *MIS Quarterly*, 9(1), 1985, pp. 77-93.

Schilling, M. "Technological Lockout: An integrative Model of the economic and strategic factors driving technology success and failure." *Academy of Management Review*, 23(2), 1998, pp. 267-284.

Schilling, M. "Winning the Standards Race: Building Installed Base and the Availability of Complementary Goods." *European Management Journal*, 17(3), 1999, pp. 265-274.

Schulte, Roy W. *Predicts 2003: SOA is Changing Software*. Gartner, AV-18-9758. December, 2002

Scott, J.E. "Facilitating Inter-Organizational Learning with Information Technology," *Journal of Management Information Systems*, 17(2), 2000, pp. 81-113

Shapiro, C., & Varian, H.R. *Information Rules*. Boston, MA: Harvard Business School Press, 1999.

Sherif, M.H. "A Framework for Standardization in Telecommunications and Information Technology." *IEEE Communications Magazine,* 39, 2001, pp. 94-100.

Smith, David Mitchell. *Web Services Standards Bodies Need to Rationalize Roles*. Gartner, COM-20-7334. August, 2003

Strauss, A. L., and Corbin, J. *Basics of Qualitative Research: Grounded Theory, Procedures and Techniques*, Newbury Park, CA: Sage Publications, 1998.

Suarez, F.F. and Utterback, J. M. "Dominant designs and the survival of firms." *Strategic Management Journal*, 16(6), 1995, pp. 415-430.

Teece, D.J. "Profiting from Technological Innovation: Implications or Integration, Collaboration, Licensing and Public Policy." *Research Policy*, 15, 1986, pp. 285-305.

Tervolin, Timo. "The Battle for Dominance: Managing the Emergence of a Dominant Design." Helsinki, Finland: Helsinki University of Technology, 2000.

Toraskar, K. V. How Managerial Users Evaluate Their Decision Support: A Grounded Theory Approach" in H.E. Nissen, H.K. Klein and R. Hischheim (eds.), *Information Systems Research: Contemporary Approaches and Emergent Traditions*, Amsterdam, Holland, 1991

Tornatzky, L.G., and Fleischeer, M. The Processes *of Technological Innovation*, Lexington Books, Lexington, MA, 1990.

Tornatzky, L.G., and Klein, R. J. "Innovation Characteristics and Innovation Adoption-Implementation: A Meta Analysis of Findings," *IEEE Transactions on Engineering Management*, 29, 1982, pp. 28-45.

Tushman M.L. and Anderson, P. "Technological Discontinuities and Organizational Environments." *Administrative Science Quarterly*, 31, 1986, pp. 439-465.

Tushman M.L. and Rosenkopf, L. "Organizational Determinants of Technological Change: Towards a Sociology of Technological Evolution." *Research in Organizational Behavior, 14,* 1992, pp. 311-347.

Utterback, J. M. and Abernathy, W. J. "A Dynamic Model of Process and Product Innovation." *Omega*, 33, 1975, pp. 639-656.

Venkatesh, V. and Davis, F. D., "A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies," *Management Science*, 46(2), 2000, pp.186-204.

W3C. [Online] The World Wide Web Consortium, Cited September 25, 2003. Available from <http://www.w3c.org>, 2003.

Watson, R. and Brancheau, J. "Key Issues in Information Systems Management: An International Perspective," *Information and Management*, 20, 1991, pp.213-223.

Welsh, Tom, analyst. *Web Services Will Succeed Where CORBA Failed.* Boston: OMG Technical Committee, 26 June, 2002. Powerpoint Slides.

WS-I. [Online] Web Services Interoperability Organization , Cited September 25, 2003. Available from <http://www.ws-i.org>, 2003.

Yoon, Y., Guimaraes, T., and O'Neal, Q. "Exploring the Factors Associated with Expert Systems Success," *MIS Quarterly*, 19(1), 1995, pp.83-106.

Zmud, R. W. "An Examination of push-pull theory applied to process innovation in knowledge work," *Management Science*, 30(6), 1984, pp. 727-738.

Zmud, R. W. "Diffusion of Modern Software Practices: Influence of Centralization and Formalization," *Management Science*, 28(12), 1982, pp. 1421 – 1431.