Florida International University
FIU Digital Commons

FIU Electronic Theses and Dissertations   University Graduate School

---

11-10-2005

Framework for Enterprise Systems Engineering

Oscar Alejandro Saenz
Florida International University

Follow this and additional works at: http://digitalcommons.fiu.edu/etd

---

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

FRAMEWORK FOR ENTERPRISE SYSTEMS ENGINEERING

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

INDUSTRIAL AND SYSTEMS ENGINEERING

by

Oscar Alejandro Saenz

2005

To: Dean Vish Prasad
  College of Engineering and Computing

The dissertation, written by Oscar Alejandro Saenz, and entitled Framework for Enterprise Systems Engineering, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

_____
Ronald Giachetti

_____
Paulette Johnson

_____
Shih-Ming Lee

_____
Martha A. Centeno, Co-Major Professor

_____
Chin-Sheng Chen, Major Professor

Date of Defense: November 10, 2005

The dissertation of Oscar Alejandro Saenz is approved.

_____
Dean Vish Prasad
College of Engineering and Computing

_____
Dean Douglas Wartzok
University Graduate School

Florida International University, 2005

DEDICATION

To God, source of love, knowledge, and wisdom

To my dear son, Oscar, a good and talented man. Thanks for giving me the joy of learning form you. Your love gives me the strength to try to be a better father.

To Eliet, insightful woman, beloved wife. Your patience and fortitude gave me inspiration through this endeavor. My endless love and gratitude for taking care of me day by day, expressing your love and support in countless details.

ACKNOWLEDGMENTS

Thanks to the members of my dissertation committee. Dr. Chin-Sheng Chen, Major Professor, for the freedom I enjoyed while conducting this research, for his words of wisdom, and for sharing his vision with me. Dr. Ronald Giachetti, for introducing me into the world of research. Dr. Paulette Johnson, for her patience in training me, I enjoyed learning the statistical side of research while working for her. Dr. Shih-Ming Lee, for believing in me since the first day we met. Dr. Duane Truex, for his contributions as Committee member during the proposal stage of this dissertation. Dr. Martha Centeno, Co-Major Professor, *mentor extraordinaire*. I was fortunate for having your advice throughout my PhD studies.

Thanks for the ongoing support of Francisco Vannini, my friend and mentor in the consulting field, a man able to create and nurture the best working environment. Thanks to EuroConsult, S.A., which granted me a partial scholarship to fund my doctoral studies.

My everlasting gratitude to my brothers and sister, my support network in the USA, for giving me so much help during my years as a PhD student. Special thanks to Eduardo J. Saenz, for his editorial advice, and to Ricardo L. Saenz, Elizabeth Saenz & Daughters, thank you for always being there for me.

Thanks to my Father, Don Oscar, always providing and caring, and Doña Cocó, loving and devoted mother, who instilled in me the passion for learning.

ABSTRACT OF THE DISSERTATION

FRAMEWORK FOR ENTERPRISE SYSTEMS ENGINEERING

by

Oscar Alejandro Saenz

Florida International University, 2005

Miami, Florida

Professor Chin-Sheng Chen, Major Professor

This research aimed at developing a research framework for the emerging field of enterprise systems engineering (ESE). The framework consists of an ESE definition, an ESE classification scheme, and an ESE process. This study views an enterprise as a system that creates value for its customers. Thus, developing the framework made use of system theory and IDEF methodologies.

This study defined ESE as an engineering discipline that develops and applies systems theory and engineering techniques to specification, analysis, design, and implementation of an enterprise for its life cycle. The proposed ESE classification scheme breaks down an enterprise system into four elements. They are work, resources, decision, and information. Each enterprise element is specified with four system facets: strategy, competency, capacity, and structure. Each element-facet combination is subject to the engineering process of specification, analysis, design, and implementation, to achieve its pre-specified performance with respect to cost, time, quality, and benefit to the enterprise.

This framework is intended for identifying research voids in the ESE discipline. It also helps to apply engineering and systems tools to this emerging field. It harnesses the relationships among various enterprise aspects and bridges the gap between engineering and management practices in an enterprise.

The proposed ESE process is generic. It consists of a hierarchy of engineering activities presented in an $IDEF_0$ model. Each activity is defined with its input, output, constraints, and mechanisms. The output of an ESE effort can be a partial or whole enterprise system design for its physical, managerial, and/or informational layers. The proposed ESE process is applicable to a new enterprise system design or an engineering change in an existing system. The long-term goal of this study aims at development of a scientific foundation for ESE research and development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION AND BACKGROUND

1.1   Introduction

Enterprises are complex systems due to the amount and variety of products they produce, the many processes, resources, and knowledge needed to make these products, and the uncertainty and relationships among all these elements (Sackett, Maxwell & Lowenthal, 1997). Such complexity is exacerbated by a modern business environment characterized by global competition, changing customer demands, technology advances, and pressure to reduce product's time-to-market and increase quality. Designing and redesigning an enterprise are complex tasks that require versatile and comprehensive methods and technologies. Many researchers have worked to develop them. This has resulted in an emerging field: Enterprise Engineering (EE). Efforts have been made to unify the language of EE for applications integration purposes (Vernadat, 2001) and to iron out the confusion among potential users caused by multiple approaches and the proliferation of multiple and heterogeneous modeling tools and languages. However, problems of little common understanding, consistent terminology, and divergent focus have persisted in the field.

This research has built upon the work done in EE and Enterprise Integration (EI) to enable greater understanding of ESE, to provide a scheme that leads to a more consistent terminology and, most significantly, to define ESE so that it has a unique and precise focus. ESE needed a comprehensive framework that specifies what enterprise systems engineering is, what the components of enterprise systems engineering are, and how enterprise systems

engineering achieves its purpose. Thus, this research has established a framework with three components: (a) an enterprise systems engineering (ESE) definition, (b) an ESE classification scheme, and (c) an ESE process.

Two distinct features of this ESE framework are that 1) it views an enterprise as a system; the system is treated as a product, and as such, the system can be designed using engineering principles, and 2) it provides a place for linking different systemic aspects of the enterprise usually addressed separately in the literature. These aspects include: strategy, for linking strategic planning with the network of enterprise elements; competency and flows, which convey coordination and the dynamic behavior of the integrated enterprise system; and capacity. The proposed ESE framework is generic; hence, applicable to any type of industry; it can support the creation of a new enterprise system or changes in an existing one.

This dissertation has been structured in seven chapters: background, research focus, literature review, definition for Enterprise Systems Engineering (ESE), classification scheme for ESE, process for ESE, and conclusions and future work. The background chapter introduces the subject and the general components of the research. The research focus shows the problem statement, objectives, and methodology. The literature review is oriented towards the understanding of ESE and gathering relevant elements from the existing literature to support the development of this research. The definition chapter offers specifications for definitions and proposes a definition for ESE. The classification scheme chapter offers specifications for classifications and proposes a classification scheme for ESE. The ESE Process proposes how to engineer an enterprise system, within the scope of the definition, based on the classification scheme and considering the interrelations among enterprise elements. Lastly,

the conclusion and future work chapter summarizes the findings of this research and their significant contribution to the ESE body of knowledge; it also includes further opportunities to extend this work.

1.2   Background

Enterprises need to adapt to the shifting environment in a constant quest for survival, stability and competitiveness (Truex, Baskerville & Klein, 1999). Enterprises adapt by improving their structure and processes, and by looking for better ways of doing what they do (e.g. implementing Total Quality Management and Business Process Reengineering initiatives) or implementing new organization models such as virtual and extended enterprises (Vernadat, 1996; ISO, 1999a). Enterprise Engineering (EE), as an emerging field of study, has the potential to support enterprises in their need for adaptation and change. Kosanke et al. (1998) argued that EE methodologies and technologies have potential for supporting an enterprise's daily operations, change management, business process integration, enterprise integration, and new organizational paradigms as extended and virtual enterprises. Similarly, Vernadat (1996) mentions that the emerging methodologies within Enterprise Engineering and Enterprise Integration (EI) are potentially powerful and useful for diagnosis of any type of flow (e.g. material, information), supporting decision-making, supporting and restructuring information systems, restructuring the organization, understanding the enterprise, designing systems, re-engineering and integrating large-scale systems, implementing Enterprise Resources Planning (ERP) systems, and managing enterprise complexity.

Pearlson (2001) stated the speed at which an organization can adapt its business processes will dictate the true competitive advantage it holds in the market. With the above potential EE can support the need for enterprise change and become instrumental for competitiveness. However, EE is still emerging and its potential to support model-based decision-making has not been yet developed (Zelm & Kosanke, 1999; Vernadat, 1996).

1.2.1   Enterprise Engineering

Although literature from the early 1970s discusses enterprise architectures in the context of enterprise engineering and integration, it was not until the 1990s that definitions of EE started to appear, establishing EE as a discipline separate from other engineering fields (Vernadat, 1996; Kosanke, Vernadat & Zelm, 1999; Vernadat, 2001).   Considering engineering as the systematic design and building of a process or an artifact by using science and mathematics (Jayachandra, 1994) and that there is consensus in viewing enterprises as open systems with a life cycle that need to work in an integrated manner, it seems logical to extend the definition of engineering to the engineering of enterprises.   However, current definitions of EE differ substantially among themselves in scope and focus.

Some definitions are broad in scope and include all aspects of the enterprise throughout its life cycle (Presley & Liles, 1996; ISO, 1999b; Presley, Sarkis, Barnett & Liles, 2001; ISEE, 2003); others view virtual enterprises and other new forms of enterprise organization as subsets of enterprise engineering (Kosanke, 1995), while others focus on business processes (Vernadat, 1996), communication networks of business processes (Kosanke et al., 1999), or an integrated set of change methods (Martin, 1995). Broad definitions counter the view of EE as a discipline at the same level of product design and manufacturing engineering

4

(Kosanke, 1995). To complicate matters, there are not only several definitions of EE, but also there are several proposals on the output of an EE process: a business process (Vernadat, 1996), a new or a modified enterprise (ISO, 1999b), an operational change (Presley & Liles, 1996; ISO, 1999b; Presley et al., 2001; ISEE, 2003), the communication networks of business processes (Kosanke et al., 1999), a changed task, business process, business unit, or entire enterprise (Martin, 1995).

Adding to the complication caused by somewhat divergent definitions of EE is that in the late 1980s, several enterprise modeling languages and almost fifty modeling tools appeared in the market targeting different enterprise elements (e.g. information and activities). Afterwards, an abundance of commercial workflow tools came, followed in the 1990s by new enterprise engineering architectures and methodologies, each presenting a different scope and process for EE. This combination brought confusion among potential users of EE, limited success of enterprise modeling methodologies, a small user community, and lack of common understanding and terminology (Kosanke, 1995; Kosanke et al., 1998; Zelm & Kosanke, 1999). An additional factor that might have contributed to the confusion surrounding the definition and output of EE is the increased importance of information systems and information systems architectures, such as the Architecture of Integrated Information Systems (ARIS) (Scheer, 1998; 1999) and the Zachman's Architecture (Zachman, 2003). Although information represents only one of the elements involved in EE, it is the focus of one recognized enterprise reference architecture as will be discussed later.

From the discussion of this section, it is reasonable to conclude that researchers have adapted their conceptualizations of EE as they gain new insights into the underlying, theoretical and

practical issues. However, much remains to be clarified and understood. A contribution of this research has been to elicit the main aspects of each of these definitions and understandings, and use this analysis as the basis for formulating a definition that better reflects our current understanding of Enterprise Systems Engineering (ESE) (see Chapter 4).

1.2.2   Enterprise Integration

One of the major deliverables of an enterprise engineering process is an integrated enterprise. Enterprise engineering is a way for achieving enterprise integration. Hence, enterprise integration is considered a subset of enterprise engineering (Li & Williams, 1994; Lim, Juster & Pennington, 1997; Bernus & Nemes, 2003; Giachetti, 2004). This is congruent with the concept of enterprise systems engineering proposed in this research. However, through the years, there has been an emphasis on the lower levels of integration, as explained in the following paragraphs.

There is substantially more literature on integration than on EE, most likely because integration has been researched longer. Early work on integration started in the 1970s with Computer Integrated Manufacturing Systems (CIM) aiming at physical interconnections between manufacturing components by means of computer networks and communication protocols (Aguilar-Savén, 2002b). Since then, several classifications of integration have been identified in Vernadat (1996), Aguilar-Savén (2002b), and Giachetti (2004):

- *Loose* (exchange of information) and *full* integration (two systems contributing to a common task, sharing the same definition of the concepts they exchange).
- *Horizontal* (physical and logical integration throughout an entire business process) and *vertical* integration (flow of decisions throughout hierarchical levels).

- *Intra-enterprise* (business processes within an enterprise) and *inter-enterprise* (among cooperating enterprises).

- Network, data, application, and business process integration. Network integration refers to physical integration of components or connectivity, e.g. connectivity of hardware, machines, devices and their operating systems. Data integration aims at data sharing, overcoming local definitions of concepts and modeling constructs. Application integration or interoperability is the ability of one application to access and use data generated by other software application. Business process integration, or enterprise integration, involves collaborating business processes and knowledge sharing to achieve coordination and goal alignment. A change of emphasis can be noticed from lower levels of integration (i.e. network, data, and application integration) in the information technology (IT) realm, towards higher levels of integration (i.e. business process integration) in the EE realm.

Business process integration has received increased attention recently to meet the needs of inter enterprise operations and coordination between enterprises (Vernadat, 1996; Kosanke et al., 1998; Kosanke et al., 1999; Ortiz, Lario & Ros, 1999; Giachetti, 2004). Coordination addresses the proper management of dependencies among activities (Malone & Crowston, 1994).

Since IT enables integration, it has led to the development of an information systems perspective of integration as a means to achieve communication and coordination, which in turn has influenced enterprise engineering. This influence is due in part to the fact that the computer science, software engineering, and information systems engineering communities

have greatly contributed to developing this field. Further, the predecessors of enterprise modeling are functional modeling and information modeling (entity-relationship and data flow models), both of which appeared in the mid-1970's in support for information system analysis and design (Vernadat, 1996).

From the IT perspective of integration, Vernadat (1996) stated that the goal of enterprise integration (EI) is the development of solutions and computer-based tools that facilitate coordination of work and information flow across organizational boundaries. Nell (1999; 2000) asserted that enterprise operations are integrated when all the processes, infrastructure, and other necessary elements can communicate the right information at the right time. Nell (1999; 2000) argued that the key for successful integration is information flow, and that enterprise integrators strive to reduce cost by computerizing information flows to make them repeatable, more accurate, and increase the speed of inter or intra enterprise communication.

Researchers have emphasized higher levels of integration and the role of EI for achieving change. Kosanke (1995) stated that lean enterprise, business re-engineering, concurrent engineering, and management of change should be viewed as subsets of enterprise integration. Similarly, Bernus & Nemes (1997) say that enterprise integration is the discipline that organizes the knowledge needed to identify the need for change in enterprises and implement that change expediently and professionally.

Lim et al. (1997) defined EI as the task of improving the performance of a complex organization by managing the interactions among the participants. EI takes into account the communication and interaction between people, organizational units, information systems,

and other resources. For Lim et al. (1997), the key enablers of EI are computer services, and the drivers of EI are business processes, information systems, facilitation of effective communication and interaction among participants, and decision-making support.

An agreement exists in the literature that the basic goals of integration are to improve overall system efficiency, responsiveness and effectiveness in the whole system compared with the isolated operation of its components, support coordination, and the achievement of the enterprise mission and goals. Hence, integration consists on the linking of the resources that perform the business processes. Resources may be people, machines, devices, applications, information systems, or computers. These links are built by means of communication networks, which again lead to seeing EI as an extension of the CIM concept towards the whole set of inter- and intra-enterprise business processes (Vernadat, 1996; Kosanke et al., 1998; Kosanke et al., 1999; Giachetti, 2004).

Referring to higher levels of integration, Martin (1995) stated that systems' performance depends more on how its parts interact than on how well they work independently of one another. Aguilar-Savén (2002b) defines EI as facilitating the task of putting together enterprise parts to form a whole in such a way that these elements together produce a better effect than the sum of their individual effects to achieve enterprise goals, which if read carefully says that EI strives to produce effective and efficient enterprise systems.

Although most authors agree that IT is the enabler of integration, Miller & Berger (2001) remarked that enterprise integration is neither an information initiative nor an information technology initiative. In other words, as stated by Kim et al. (2003), EE projects have a

broader scope than software engineering projects. Moreover, some authors emphasize that computerization is not an objective by itself unless it supports integration in the business and enterprise sense (Ortiz et al., 1999). In lieu of this, most researchers recognize the importance of higher levels of integration, e.g. business process and enterprise integration (Li & Williams, 1994; Aguilar-Savén, 2002b; Vernadat, 1996; Kosanke et al., 1998; Kosanke et al., 1999).

This section has demonstrated that most authors view EI as a way to improve overall efficiency and effectiveness, while others lean towards IT and emphasize how to achieve integration through IT. Nevertheless, weaved within their views is the concept that enterprise integration is a subset of enterprise engineering. While this research considers physical, data, and application integration, it strives to support coordination in the whole enterprise system as will be presented later.

1.2.3   Enterprise Architectures

Enterprises can be considered as a final product, which entails that they are the final deliverable of a process. During the life cycle of this process enterprises have to be analyzed, designed, built, and put into operation (Bernus & Nemes, 1996). Enterprise architectures provide theoretical support for such a process. This research extends the scope of existing approaches and has produced a classification scheme integrated with a robust definition and a process for ESE.

Enterprise reference architectures and their associated methodologies represent the main efforts towards EE. They are the main source of information in regard to modeling

approaches and processes for EE. Enterprise reference architectures enable consistent modeling of the enterprise. Enterprise modeling is a way to structure and manage enterprise complexity by decomposition and a way to describe functionality and behavior of the operation (Zelm & Kosanke, 2001). Enterprise *reference* architectures are high level enterprise models, or meta-models for a set of enterprise models. They attempt to describe the steps to develop an enterprise and the structure and relationships of these steps throughout the life cycle of an enterprise. Enterprise models can be used for documentation, analysis, re-design, and operation of a company. An enterprise model may represent what an enterprise does and how it operates. The relevant parts and the level of detail of an enterprise model depend on their area of concern and intended use (Reithofer & Naeger, 1997; Williams, 1998; Williams, Li, Bernus, Uppington & Nemes, 1998; ISO, 1999b; Kosanke & Nell, 1999b; Zachman, 1999; Bernus & Nemes, 2003; Aguilar-Savén, 2004). In general, enterprise architectures strive to understand business processes, which is the first step towards analysis and redesign (Luo & Tung, 1999).

Enterprise systems are complex; hence, it is widely accepted that different life-cycle phases can be represented by different models. Enterprise reference architecture's views help represent an enterprise as a whole, which promotes understanding, acceptance, and reduction of modeling complexity. Enterprise reference architectures have common characteristics: multiple views of the enterprise, taxonomy of concepts, common language, attention to life cycle, and attempt to represent the relationships (i.e. information exchange) between the life cycle phases of an enterprise (Bernus & Nemes, 1996; 2003).

11

Some enterprise architectures, like the Computer Integrated Manufacturing Open Systems Architecture (CIMOSA), do not limit the number of views to consider. A variety of views have been proposed. Miller & Berger (2001) considered the business view of an enterprise (i.e. market analysis, product concept, development program launch, and customer satisfaction) to be the dominant view; it is the one that addresses customers and markets, and drives the enterprise to respond to customers and a competitive environment. Miller & Berger (2001) argued that the impact of change in an enterprise can be analyzed from high level to lower level architectures. High-level architectures are the enterprise strategy and business architecture, followed by processes, resources (first physical and then human resources), and the lower level IT architecture.

Whitman et al. (2001) stated that there is no single universally correct architecture. Until the early 1990s, there were only three major enterprise architectures in the literature: The Purdue Enterprise Reference Architecture (PERA), the GRAI Integrated Methodology (GIM), and the Computer Integrated Manufacturing Open Systems Architecture (CIMOSA) (Williams, 1998). Aguilar-Savén (2002b) concluded that these were still the main reference architectures. The newer Generalized Reference Architecture and Methodology (GERAM) merges aspects from CIMOSA, PERA, and GIM. All of these adopt a holistic approach however PERA adopts a resource perspective; GIM focuses on the decision system; and CIMOSA tends to focus on the representation using its own language for future computerization. A special case would be GERAM, which attempts to provide a standard for new enterprise architectures. These architectures are presented in detail in the next chapter.

Beside the reference architectures previously mentioned, this research reviewed two additional architectures. Although they are specifically oriented toward information systems they provided a fresh insight into enterprise systems in the context of this research. They are: ARIS, which has been used in reengineering projects and has been considered to be one of the market leaders in enterprise modeling (Reithofer & Naeger, 1997), and the Zachman's architecture, which has been recently modified and relabeled as an enterprise architecture. The enterprise reference architectures represent significant advances in enterprise systems design. However, these architectures do not encapsulate a fundamental and theoretical design philosophy. According to Grunninger (2003), one persisting problem is that enterprise design has been descriptive and ad-hoc, when indeed what is needed and desirable is to define a theory of enterprise design and its underlying principles. This research contributes a systematic approach that better explains the underlying complexities to *engineer* enterprise systems.

1.2.4 Enterprise Engineering and Strategy

Porter (1996) regards strategy as the creation of a unique and valuable position, which involves the design of a different set of activities for competing, making trade-offs, and creating fit among activities so that they reinforce one another. Strategy imposes constrains for engineering an integrated enterprise system, for coordinating different enterprise elements among themselves, and for aligning those enterprise elements with the enterprise goals in order to face environmental challenges. Building an integrated enterprise is intrinsically linked to strategy, particularly strategy implementation given that, according to Kaplan & Norton (2000), the ability to execute strategy is more important than the strategy itself.

Strategy can be divided in three levels: corporate, competitive (or business), and operational (Gaither & Fraizer, 1999; Coulter, 2002). Corporate strategy defines what the business is, its objectives, results, customers, and what a customer values and pays for. It decides how to allocate funds, e.g. which project's budget to increase, decrease, or terminate, and what will be acquired or outsourced. Corporate strategy defines the destination of the enterprise, the best way to get there, in what value chain to compete, and the position of the business within that value chain. The term value chain gained relevance because the total cost is what matters, independently of who holds ownership of a part of the total process. In short, corporate strategy is concerned with the commitment of present resources to future expectations, focusing on the long-term and selecting a business the company wants to compete in. Competitive or business strategy identifies the optimal value the enterprise wishes to deliver, and it defines how an enterprise is going to compete in a specific business or industry. Operational strategies focus on specific functions, such as marketing, manufacturing, finance, and e-Business. They are formulated for the short-term, although, some operational aspects may have long-term impact, such as facilities capacity, location, and technology (Drucker, 1999; Manganelli & Hagen, 2003).

Ortiz et al. (1999) stressed the importance of keeping business operations aligned with strategy, and stated that each process must have defined objectives and support the enterprise strategy. Similarly, Kettinger & Teng (1998) said that a business process must be aligned with strategy, people, structure, and IT. Alignment is a term commonly found in the business and management literature and is used to signal the need for a match or fit among enterprise components such as resources, business process, and strategy. Manganelli and Hagen (2003) explained alignment as having a shared mission, vision, issues, challenges,

goals, core values, operating principles and how to close the gap between the current and the target future state of the enterprise.

Vernadat (1996) stated that the global economy forces companies to realign not only their business processes but also their organizational structure, suggesting that alignment has to do with making the necessary changes within an enterprise to cope with competitive requirements or changes in the environment. Luo & Tung (1999) said that each business process in an enterprise must have well defined objectives and outcomes. According to Watkins (1997) any product, service, or project is strategically aligned if it contributes to the enterprise objectives (i.e. corporate, division, and business-unit objectives). Objectives, a central element in business and manufacturing strategy, play a role as a control element in the hierarchy of business process (Chandra & Kumar, 2001).

The process of formulating, implementing, and evaluating strategies is called strategic management. It includes defining a mission; external and internal auditing; formulating long-term and annual objectives, formulating strategies and policies, and implementation (David, 1997; Coulter, 2002). Kotler & Armstrong (2001) emphasized that strategic management is the process of developing and maintaining a strategic fit between the organization's goals and capabilities and its changing environment.

Relationship between strategy and the engineering of an enterprise system is well-cited in the literature. Operational strategies are based on properties that can be designed into the enterprise system. It is the domain of *strategy* to decide what design properties (e.g. reconfiguration capability, flexibility) an enterprise system must exhibit; to what level the

system requires those properties, and how to best incorporate those properties into the system (Giachetti, Martinez, Sáenz & Chen, 2003). When EE defines and designs enterprise systems it is simultaneously defining and designing the properties that are responsible for achieving certain operational performance.

Certain aspects of strategy are included in enterprise architectures, such as PERA, CIMOSA and GERAM, because engineering an enterprise system must be done within the boundaries set by a vision. Strategic vision drives overall enterprise engineering and specific value streams (Martin, 1995). Enterprise success requires both an effective strategy and adaptability to the environment (David, 1997). However, a specific framework that addresses how to engineer an enterprise system and link it with strategy has not yet been developed.

CHAPTER II

RESEARCH FOCUS

2.1 Problem Statement

Several general problems have been identified in the emerging field of enterprise engineering (Kosanke et al., 1998; Zelm & Kosanke, 1999; Vernadat, 2002). Efforts have been made to unify the language of EE for applications integration purposes (Vernadat, 2001) and reduce the confusion among potential users caused by multiple approaches and the proliferation of multiple, heterogeneous modeling tools and languages; however, problems on little common understanding, consistent terminology and divergent focus persist in the field. There is lack of business justification, little management involvement, and little use of existing enterprise engineering architectures. There is a small user community due to little awareness of EE; and there is a tendency on the part of small and medium enterprises to ignore enterprise modeling and enterprise integration. The above general problems are seen as an effect; this research addresses some of their causes, targeting three areas: (a) an enterprise systems engineering (ESE) definition, (b) a classification scheme for ESE; (c) an ESE process.

In regards to a *definition of ESE*, it is clear that the existence of several definitions of enterprise engineering with different foci has been at the center of the above mentioned general problems. While some definitions of EE are rather broad (ISO, 1999b; ISEE, 2003), others focus on change methods (Martin, 1995), and yet others focus on business processes (Vernadat, 1996) or communication networks and life cycle (Kosanke et al., 1999). Broad definitions with different foci do not portray the uniqueness of EE as a separate research field

17

and do not help to orchestrate efforts toward the development of EE. According to Rowe, Truex, & Kvasny (2004), a field of study must have a central character and distinctiveness. Current definitions of EE may have a central character, namely, the enterprise, but they do not have distinctiveness. Divergent foci do not support the concentration of efforts toward the development of EE; instead they contribute to the existing confusion among potential users (Kosanke et al., 1998; Zelm & Kosanke, 1999). Thus, it is necessary to continue the efforts towards properly defining enterprise systems engineering.

In regards to an ESE *classification scheme*, the main sources of frameworks and methodologies for enterprise systems engineering are the enterprise architectures (CIMOSA, PERA, GIM, and GERAM). Recognized enterprise architectures do not fit some definitions of enterprise engineering: CIMOSA focuses on building an information system through its own language; GIM focuses on the decision system and does not include implementation; PERA's Master Plan (of 300+ pages) presents a process to design an integrated enterprise focusing on life cycle and resources; GERAM originated from the merging of the other three enterprise architectures (CIMOSA, PERA and GIM) and it does not have its own process. All these enterprise architectures attempt to reduce complexity by modeling and by providing general representations of the relationships among different enterprise views and abstraction levels during the life cycle of an enterprise (Kosanke, 1995; Vernadat, 1996; Chen, Vallespir & Doumeingts, 1997; Williams & Li, 1998; Kosanke et al., 1999; Kosanke & Zelm, 1999; Williams, 1999). However, these enterprise architectures are still complex, which makes them less attractive to business users (Noran, 2003).

An enterprise is composed of various systems, which are expected to interact cohesively to achieve the enterprise goals. Thus, an enterprise is a system in its own right and engineering principles should be applicable to its design. Enterprise architectures are only starting to highlight the different areas of study within the enterprise that need to be addressed to produce the desired output, which is an integrated enterprise system. A single graphical representation, used by existing enterprise architectures, is not able to encompass most of the areas that need to be addressed to engineer an enterprise system.

In regards to an ESE *process*, it is clear that enterprise architectures are intended to support the design of an integrated enterprise system through a process or methodology. Without a process the architecture achieves nothing. Williams et al. (1996) stated that an enterprise methodology is more important than the architecture itself. Similarly, Tolle & Vesterager (2003) stated that in the context of virtual enterprises, a methodology is needed that helps manage the task of creating an enterprise. Although there is agreement regarding the need for designing an integrated enterprise system, the problem is that several choices have been suggested regarding what the output of an EE process should be. Among the suggestions are a business process (Vernadat, 1996), a modified enterprise (ISO, 1999), implementation of an enterprise element (ISEE, 2003), communication networks (Kosanke et al., 1999), and a changed task or a changed enterprise (Martin, 1995). In fact, the existence of several proposals for the output of an EE process impedes EE from becoming a distinct discipline. Different choices of output lead to different EE processes to produce that output. Moreover, the variety of EE processes and outputs will continue setting the stage for increased modeling approaches and tools.

A common thread in most enterprise architectures is the significance given to integrating strategy in an EE process (Scheer, 1998; Williams, 1998; Kosanke & Zelm, 1999; Scheer, 1999; Veasey, 2001; Zachman, 2003). Similar importance is given in the literature to the subject of alignment among business processes. Business processes must be aligned among themselves and with strategy (Ortiz et al., 1999). At its current development enterprise engineering methodologies signal the need to integrate strategy, but none indicate their relationships with different levels of strategy during the engineering of an enterprise system. This constrains management involvement and business justification of EE (Kosanke et al., 1998; Zelm & Kosanke, 1999).

There is a need for extensive research in EE if it is to grow as a field and become the source of concepts, methodologies and tools to design, improve, and redesign enterprises of the 21st century. The proposed framework intertwines a definition, a classification scheme, and a process for engineering integrated enterprise systems. It is an important step toward overcoming significant challenges faced by today's EE community.

Enterprise Systems Engineering needs a framework that:

- Clearly defines what ESE is.

- Has a process to engineer an integrated enterprise.

- Sets the boundaries of EE with respect to other disciplines.

- Organizes the different areas of study that ESE needs to address.

- Enables the use of engineering principles and methods to produce an enterprise system.

The lack of such a framework, together with the existence of several enterprise architectures, each one with its respective methodology attempting to fill the void, has contributed to curtailing the use and spread of EE methodologies (Kosanke et al., 1998; Zelm & Kosanke, 1999).

2.2   Research Objectives

The goal of this research has been to develop a framework for enterprise systems engineering, which guides the engineering of an enterprise throughout its life cycle, systematically and cohesively.

The framework was initially conceived as having three consistent components: a definition, a classification scheme (which will be the base for the ESE process and its scope), and an ESE process. The acronym ESE (Enterprise Systems Engineering) will be used instead EE to highlight that an integrated enterprise – the end-result of an ESE process – is seen as a system of systems.

To achieve the goal, a set specific objectives and deliverables were established as follows:

A. *Development of an ESE definition*. This definition answers the question: What is enterprise systems engineering? The definition includes the elements of business processes, because integration of enterprise components depend on the integration of business processes (Vernadat, 1996). The definition is intended to distinguish ESE from other engineering fields. The purpose of including this objective as part of the

deliverables of this research has been the need for a consistent ESE framework, which starts with an understanding of what ESE is.

B. *Development of an ESE classification scheme*. This is a graphical representation that answers the question: How can a single model show all the areas that need to be addressed to engineer an enterprise system? The classification scheme provides a notation to identify ESE areas and to classify research efforts.

C. *Development of an ESE process model*. The ESE process answers the question: What needs to be done to design an integrated enterprise system? The process guides the design of an enterprise system, which is its final product. This process considers strategy and specifies what level(s) of strategy must be incorporated. The ESE process is centered on engineering the enterprise elements that are at the core of an enterprise system.

2.3   Research Methodology

The research methodology used qualitative research methods, incorporating both inductive and deductive reasoning. Inductive reasoning strives to develop generalizations based on a limited number of observations, as opposed to deductive reasoning that allows for the development of specific predictions based on general principles or observations. Qualitative research uses inductive reasoning to analyze information interpretively by organizing data into categories, identifying patterns, and producing a descriptive narrative synthesis. Categories or dimensions of analyses emerge as the understanding of the subject under investigation grows. Qualitative research is useful for exploring in depth and detail complex

and little known research areas, which is the case of ESE (Gay & Airasian, 2000; Patton, 2002). Qualitative research in enterprise integration was used by Aguilar-Savén (2002b). Aquilar-Savén used both, empirical information from cases studies to infer conclusions – inductive approach – and a deductive approach, starting with a review of previous research before contrasting it with empirical data. In general, recognized enterprise architectures have been formulated in a similar way and later tested into practice.

Specifically, this research follows closely two qualitative research methods: comparative analysis and the negative case analysis. Comparative analysis is used to examine the literature, identify concepts and categories, and look for distinctive characteristics for understanding and explaining. Negative case examples and discrepant or contradictory evidence to challenge emerging concepts were used to disconfirm, change parts, or alter the scope of early versions of the ESE framework (Gay & Airasian, 2000; Patton, 2002).

One of the difficulties in applying research methods based on observation, interviews, and other qualitative approaches is that they rely on the interpretive skills of the researcher to analyze, integrate, and make sense of the data collected. Qualitative research posits that meaning is dependent on perspective or context. Individuals and groups have differing outlooks, interests, biases, foci, and experiences, all contingent on cultural and environmental contexts as well as personal world views. No single perspective is necessarily more valid than another (Gay & Airasian, 2000). This is one reason that different approaches towards enterprise engineering exist; people in varying world regions developed distinct enterprise architectures (CIMOSA and GIM-GRAI in Europe and PERA in America), with different

orientations (the information system in CIMOSA, the decision system in GIM, the physical system in PERA).

This research follows a deductive approach to build a theoretical foundation as starting point. Then, it has used the inductive approach, comparative analysis and the negative case analysis, to develop a general framework from existing literature on enterprise engineering and from empirical experiences in creating, designing, and improving enterprises. Specific theories and methods used to support this research included:

- The IDEF0 (Integrated Computer Aided Manufacturing Definition) methodology for enterprise modeling, which is used to create activity models and establish interrelationships among inputs, outputs, mechanisms and controls in the modeling of an ESE process. The IDEF1x methodology, which is used to develop metamodels where needed to specify relationships among the concepts developed or used. Petri Nets, specifically place-transition nets, which are used to describe concurrency in the ESE process. Product design and development theory, specifically the product development process of Ulrich and Eppinger (2000) and the theory on axiomatic design by Suh (2001), which are used to support the design of the enterprise engineering process and to convey requirements and other critical factors as a way to check alignment among the enterprise components. PERA, CIMOSA, and GIM have been used as benchmarks for the proposed classification scheme and ESE process and for validation purposes.

2.4   Research Scope and Assumptions

All proponents of Enterprise Engineering visualize it as a separate discipline, even at the same level of product and manufacturing engineering (Martin, 1995; Presley & Liles, 1996;

Vernadat, 1996; ISO, 1999b; Presley et al., 2001; ISEE, 2003). This research is consistent with the research community agreement and recognizes enterprise systems engineering as its own discipline, distinct from industrial engineering, manufacturing engineering, product engineering, software or ERP systems engineering.

The *scope* of the ESE framework is generic, that is, applicable to any type of industry as others have proposed (CIMOSA, GIM, GERAM, PERA). There are two main reasons for a generic framework. First, after a extensive survey of leading industries, Manganelli and Hagen (2003) found that the basic nature of businesses, best business practices, and subsequent major problems that industries face have not changed in more than twenty years. The management of systems is still the main difficulty: value is created by the system, not by its parts. Second, Druker (1999) affirmed that 90% of what organizations are concerned with is generic and only 10% has to be customized to the organization's specific mission, culture, history and vocabulary. Furthermore, the differences resulting from the latter 10% are no greater between businesses and non-businesses.

While recognizing the importance of the soft aspects of an enterprise system, this research does not directly address them. Culture (principles, policies, attitudes, and the social side of the enterprise), and acceptance or resistance to change are fundamental aspect for the enterprise success (Molina, 2003). The ESE framework does not attempt to engineer the enterprise culture. Abundant sources of information exist for the subjects of human resources and management; therefore, they are out of the scope of this research.

This research does not delve into project management and other support activities that are commonly implemented throughout the engineering activities (Nalbone, Vizdos & Ambler, 2004), such as modeling for specific areas and change management.

GERA mentions that while one enterprise is subject to change, other enterprises may be responsible for the formulation of its strategy, its construction, or the implementation of a project to change it (Bernus & Nemes, 1996; 1997). This framework does not focus on who is in charge of formulating or implementing strategy. Rather, the focus is having a strategy for building competencies needed to compete in future markets and creating a blueprint that guides the integration of the whole enterprise system (Kalpic, Pandza & Bernus, 2003).

Assumptions for this research are based on the idea that if enterprises can be viewed as products, they can be designed, built, and put into operation (Bernus & Nemes, 1996). Product design theory can be also used to support the design of processes, systems, software, organizations, manufacturing systems, and business plans; the design process is all the same at some conceptual level and it can be used in different disciplines. Thus, another assumption is that product design can be extended for enterprise systems design (Suh, 2001).

CHAPTER III

LITERATURE REVIEW

An objective of this effort has been to clearly define what ESE should be at a time when EE is in its infancy. Therefore, it has been necessary not only to review the works that have led to the birth of EE but also to review how to formulate definitions based on the philosophy of science literature. Consequently, this chapter presents two threads: philosophical principles to formulate definitions and technical works on EE. The emergent body of literature in ESE is organized in three categories: enterprise systems; enterprise frameworks and architectures; and enterprise strategy focusing on the potential relationships with ESE. Designing an ESE framework required knowledge from other areas, including IDEF methodology, Systems Notations, Product Design, and Petri Nets.

3.1   Definition

Defining is a basic philosophical activity, and as Xia (1999) asserted, a clear definition of objects under investigation is of prime importance in science. Without a clear understanding of the subject of inquiry from the beginning scientific research cannot take place (Chakrabarti, 1995). Moreover, definitions are abstractions that separate an object from the rest of the world in a way that gives new knowledge of the object (Robinson, 1968).

3.1.1   Types of Definition

There are several types of definition (Robinson, 1968; Copi, 1982; Copi & Burguess-Jackson, 1995; Xia, 1999). However, only four types are related to this research:

- Stipulative or nominal.

- Lexical.

- Precising.

- Theoretical.

Stipulative, or nominal definition, is used to assign meaning to a new term, symbol, or name. It sets up the meaning and relationship between a word and an object represented by the word. It is a request to use the definiendum to signify what is meant by the definiens. Stipulative definition is useful for parsimony in written reports to remove ambiguity, and to improve or create new concepts. *Lexical* definition is used for terms that have an established usage; it documents the existing meaning of a term, increases vocabulary, or eliminates ambiguity. *Precising* definition is used to further explain a term when it is vague. *Theoretical* definition is used to propose a scientifically useful description of the objects to which the term applies; therefore, it is a statement of the essential nature of an object.

3.1.2   General Purposes of a Definition

A definition explains what the definiendum is (Chakrabarti, 1995). Copi (1982) and later Copi and Burguess-Jackson (1995) presented four purposes of a definition:

- Increase vocabulary.

- Eliminate ambiguity.

- Reduce vagueness.

- Formulate scientifically.

Increasing vocabulary and influencing attitudes are purposes out of the scope of this research but they are mentioned for completeness. *Eliminating ambiguity* is necessary as when a

28

word have one of two or more distinct meanings in the same context. The purpose of eliminating ambiguity is particularly relevant to this research because the ordered set of words "enterprise systems engineering" have a different meaning than the aggregation of its components words. *Reducing vagueness* is necessary when a word refers to range of variation in quantity, number, or intensity. Vagueness is reduced by clarifying the applicability of a term in a given context. This purpose becomes relevant in this research because the term 'Enterprise Engineering' has been given several different meanings varying in scope and focus; therefore, it is necessary to introduce clarity. A *scientific formulation* is necessary when assigning meaning to the term being defined based on the most useful or relevant characteristic.

3.1.3   Techniques for Defining

Copi (1982) and later reinforced by Copi & Burgess-Jackson (1995) mentioned five techniques for defining:

- Denotative.

- Synonymous.

- Operational.

- Synthesis.

- Genus and difference.

Denotative defines by extension. It gives examples as in a complete or partial enumeration of objects defined by the term. A special case of this technique is the *ostensive*, which uses gestures to show the objects referred by the term being defined. *Synonymous* uses another word which has the same meaning. *Operational* defines based on a set criteria (Hempel,

29

1965; Copi, 1982). *Synthesis* assigns meaning using the relationships of an object to other objects in a whole, or by how the meaning arises or by how it is caused. Robinson (1968) states that the definition of a concept often takes the form of synthesis, specifying its place in a larger system of concepts or expressing it in terms of other primitive concepts. *Genus and difference* defines by division, by analysis, or by connotation. It is broadly used in biology to group organisms into categories.   A term is defined by naming a genus (i.e. a class). The term being defined is a subclass of the genus, so the characteristics that differentiate the term from other terms within the genus are specified. Definition by genus and difference is applicable to terms that have complex attributes but it cannot be applied to terms connoting universal attributes because there is no broader genus for them (Chakrabarti, 1995).

3.2   Enterprises and Systems

The understanding of what an enterprise is and what a system is supports the formulation of a definition and scope for ESE. Enterprises focus on their customers and on responding effectively to changing customer needs (Kotler, 1994). An enterprise may be a for-profit or a non-profit organization. Vernadat (1996) stated that an enterprise can be viewed as a large set of concurrent processes executed by communicating agents. According to the ISO standard ISO15704 (1999b) "Requirements for Enterprise-reference Architectures and Methodologies", an enterprise is a group of organizations sharing a definite mission, goals, and objectives to offer an output such as a product or service. A related view is offered by Presley et al. (2001), who stated that an enterprise is a collection of activities organized into a set of business processes that cooperate to produce desired results. They defined activity as any organized behavior that transforms inputs into outputs.

Another view of an enterprise that directly influences its engineering is provided by Bernus & Nemes (1996, 2003), who argued that enterprises may be viewed as products that need to be invented, specified, designed, built, and put into operation. Viewing an enterprise as a product is valid for new and existing enterprises; the latter may be considered as an existing product suitable for redesign. The same authors stressed that the theories, tools, methodologies, and activities used to engineer an enterprise should be applicable without regard to the nature of the business, a perspective that underscores this dissertation. The conception of an enterprise as a product implies that an enterprise may be considered a deliverable of a process, specifically an ESE process.

Other views emphasize that enterprises are systems. Enterprises are dynamic, purposive, and densely connected systems (Checkland, 1982). An enterprise is a collection of processes, technology, and people working as a system (Kosanke & Nell, 1999a). The International Organization for Standardization (ISO), a worldwide federation of national standards bodies, published the International Standard ISO14258 regarding concepts and rules for computer-understandable enterprise models to facilitate process interoperation. According to this standard, an enterprise is a system, and it and its models must conform to system theory (ISO, 1999).

Regarding the concept of system, Hanson (1995) defines a *system* as any two or more parts that are related, such that a change in any one part changes all parts. The interdependencies among the parts define the structure of the system, which cause the properties of the whole to be different from the concatenation of properties of the constituent elements (ISO, 1999a). Similarly, Wilson (1984) envisioned a system as a set of components linked together to

achieve some purpose. A system has a hierarchy, it has subsystems within it and at the same time the system is a subsystem of a wider system. The description of lower levels in the hierarchy provides details on how the system performs and achieves its purpose, whereas the description of higher levels show the role of the system in its environment (ISO, 1999a).

Checkland (1982) said that a system is characterized by (1) its hierarchical structure, where smaller entities are themselves wholes; (2) its emergent properties, attributed to the whole not to the parts; and (3) its control, which provides a mechanism by means of which the system adjusts itself to continue pursuing its purpose based on some performance measurement. A system can also be described in terms of its customers, its transformation, the actors who perform the transformation, a world view that guides the decision making, the owner of the system, and environmental constraints (Checkland & Scholes, 1990).

Business processes have been considered to be at the core of an enterprise. They play a significant role in contemporary organizational and operational paradigms and in the evaluation of business partners. There are several similar definitions of business processes, all of them make reference to sequence of activities, but some go beyond and classify business processes, or mention their component activities or the resources that perform them. The importance of business process for enterprise systems engineering is explained in terms of their comprehensiveness. Business processes bring with them the flow of material, information, control, and resources that perform them; therefore the integration of other enterprise components depend on the integration of business processes.

A business process is defined as a sequence of enterprise activities that cross the boundaries of organizational units and whose execution is triggered by some event and produces an observable or quantifiable result for a defined customer. In regards to their components, business processes are made up of material and information processes (Davenport & Short, 1990; Martin, 1995; Georgakopoulos, Hornick & Sheth, 1995; Vernadat, 1996; ISO, 1999; Kosanke, 1999; Appelrath & Ritter, 2000; Kotler & Armstrong, 2001).

When engineering a business process that spans two or more enterprises, a supply chain arises. A supply chain (SC) is a network of autonomous enterprises solving a common problem. These autonomous entities have interacting physical inputs/outputs and collaborate to sustain the progress of the individual entities and of the network as a whole. Chandra and Kumar (2001) reported that during the 1990s strategic alliances, motivated by global competition and common objectives, forced enterprises to focus on the total cost from source to consumption. SC encompasses IT integration and coordination of planning and control of all activities aimed at producing and delivering a product from the supplier's supplier to the customer's customer (Lummus, Krumwiede & Vokurka, 2001).

Information Technology has enabled new forms of enterprises such as virtual and extended enterprises, allowing for autonomy and distribution of responsibility and power (Mukherji, 2002). A *virtual* enterprise is a temporary alliance of several distributed, autonomous, product-oriented work units, manufacturing a particular product in order to meet a market need rapidly. These virtual partnerships will form, operate and dissolve quickly, and will demand more support from current enterprise models (Reithofer & Naeger, 1997). A virtual enterprise is an enterprise in which all the aspects of a traditional enterprise apply, but in

addition time plays an important role given the rapid formation and dissolution of this type of enterprise (Nell, 1999). An *extended* enterprise refers to a more permanent relationship among one organization and its customers, suppliers and trading partners (Martin, 1995; Simchi-Levi, Kaminsky & Simchi-Levi, 2000).

Williams et al. (1996) stated an enterprise system may consist of a part of a business unit, several business units, or the whole enterprise. Following this trend of thought, an enterprise is therefore a system made up of a set of business processes that share a common mission and objectives. An enterprise system may be a part of a business process, a whole business process, a set of business processes, companies working independently or as part of a partnership (as in a supply chain), or a virtual or extended enterprise.

3.3   Enterprise Frameworks

A framework contains a set of architectural representations. A framework provides a way for better comprehension and communication of architectural concepts and their specifications. They also facilitate the improvement of development methodologies and tools oriented towards integration. Each architectural representation models part of a system, its components and interactions, and it can be further linked to its own methodologies and tools (Zachman, 1999).

The literature has numerous articles addressing enterprise frameworks. The newness of the field is readily observable by the fact that authors go back and forth between the term "framework" and the term "architecture". According to Zachman (1999), virtually all the reference architectures and information systems architectures summarized in this section

classify as frameworks because they allow the use of multiple representations of the enterprise. There are also articles discussing enterprise models; however, several of these models are included in other architectures or they do not present evolution over life cycle. Among these are: the Structured Analysis and Design Technique (SADT); the Integrated Enterprise Modeling (Reithofer & Naeger, 1997), the Totally Integrated Enterprise framework of Miller and Berger (2001), the CEN-ENV40-003 (Vernadat, 1996), and the framework proposed by Wu and Ellis (2000) in the context of manufacturing systems design.

The following sections offer a succinct, contextual review of the GRAI Integrated Methodology (GIM); the Computer Integrated Manufacturing Open Systems Architecture (CIMOSA); the Purdue Enterprise Reference Architecture (PERA); and the Generalized Reference Architecture and Methodology (GERAM), which resulted from the mergence of the previous three. Two information systems-oriented architectures are also presented in this section: Zachman's and ARIS.

3.3.1   GRAI Integrated Methodology

The GRAI Integrated Methodology (GIM) is one of the earliest efforts in ESE. It was started by the GRAI Laboratory of the University of Bordeaux, France, in 1974 (Williams, 1998). GIM is focused on the design of a new system. Its life cycle does not include construction and operations (Williams & Li, 1998). GIM stresses the link between organizational structure and the decisional system (Zülch, Rinn & Strate, 2001). GRAI stands for *Graphes à Résultats et Activités Interreliés* (Graphs with Interrelated Results and Activities). The salient feature of GIM is that it allows the modeling of the decisional structure of an

enterprise system whereas other existing architectures tend to focus on the information system (Li & Williams, 1994).

The elements of GIM are as follows (Chen et al., 1997):

- GRAI conceptual model: a representation of basic concepts of a manufacturing system with three sub-systems: decision, information, and physical.

- The GIM modeling framework, which has two dimensions: views and abstraction levels. The four views are information, decision, functional, and physical. The abstraction levels are: conceptual, structural, and realizational. Each combination of abstraction levels and views results in a different sub-model of an enterprise.

- The GIM structured approach, which has a life cycle that includes three phases: analysis, user oriented design, and technical oriented design, with an initialization node and an implementation node. The information, decision, functional, and physical views are addressed through this life cycle

- GIM modeling formalisms: the two basic modeling formalisms are the GRAI grid and the GRAI nets. The GRAI grid is used to perform top-down analysis in the form of a matrix of functions, decision levels, and decision horizon. The GRAI nets are used to perform bottom-up analysis in terms of activities, resources, and input/output objects. It is used to build a decision system model (Vernadat, 1996).

- GIM case tool: PROGRAI is a tool that supports the use of GIM.

The GRAI conceptual model is shown in Figure 1 and the structured approached of its methodology is shown in Figure 2. These figures clearly indicate that information is just one part of the architecture.
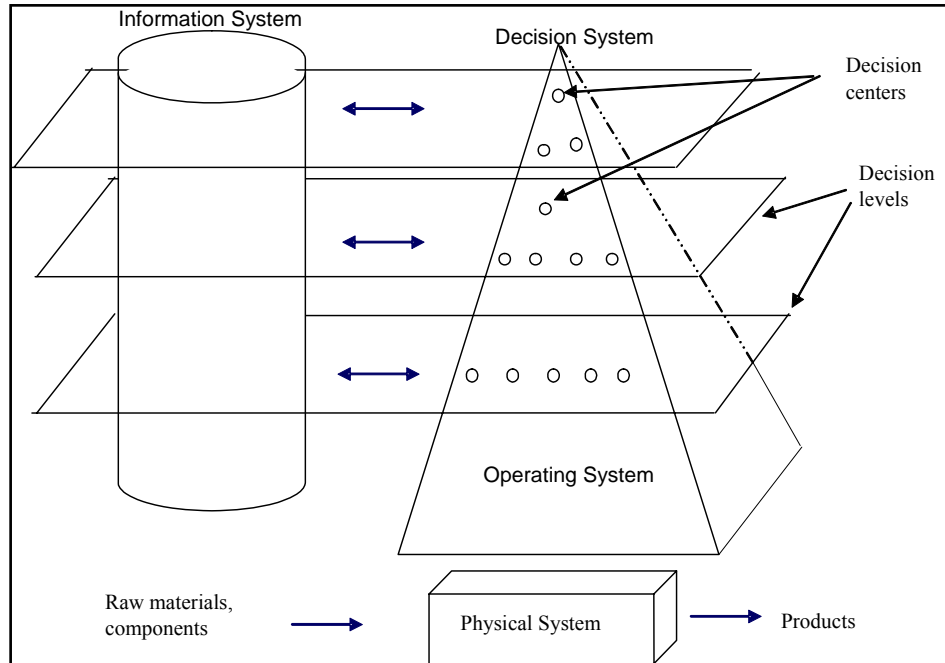
Figure 1: GRAI Conceptual Model (Vernadat, 1996)
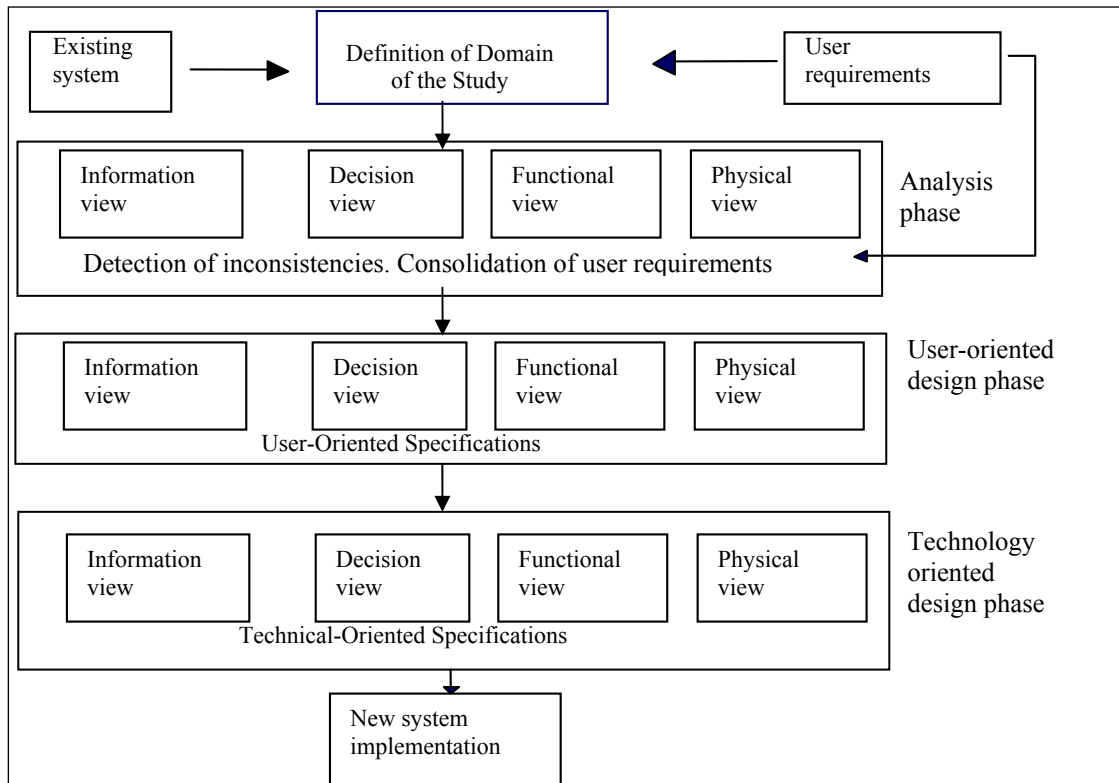
Figure 2: GIM Structured Approach (Vernadat, 1996)

3.3.2   Open Systems Architecture for Computer Integrated Manufacturing

Efforts on the CIM Open Systems Architecture, or Open Systems Architecture for Computer

Integrated Manufacturing (CIMOSA), started in 1984. It was developed by the European

CIM Architecture Consortium (AMICE), backed by the European Community (Williams,

1998). The CIMOSA Association is a non-profit organization involved in promotion of

Enterprise Engineering and Integration based on the CIMOSA architecture and methodology.

The goal of CIMOSA is to establish standards, with emphasis on a framework for enterprise

modeling, an enterprise modeling language, and an integrating infrastructure for model

enactment, all supported by a common terminology (CIMOSA, 2003).

CIMOSA considers an enterprise as a large collection of concurrent processes and interacting

agents that perform the processes. These processes are intended to achieve some business

goal and are executed upon request (Berio & Vernadat, 1999). CIMOSA covers four

enterprise views: function, information, resource, and organization, but it does not constrain

the number of views that can be added to the framework. CIMOSA has a life cycle of three

phases: requirements definition, design specification, and implementation description. It also

has three genericity levels: generic, partial, and particular (Kosanke et al., 1999; Vernadat,

2001).

The *function* view describes business processes and functionality using activities and

behavioral rules. Behavioral rules specify conditions under which activities may start.

Activity's inputs and outputs are described using enterprise objects and information elements

constructs (Sternemann & Zelm, 1998). The *information* view lists the information required

by each function, and how it is collected, handled and stored. The *resource* view describes

38

the resources responsible for the execution of tasks in terms of their capabilities, capacities, costs, and their relationship to the functional and control structures, and to the organizational structure. The *organization* view is a description of the responsibilities assigned to individual resources of the enterprise for operation and control of the enterprise and their relationships to each other (Li & Williams, 1994). The CIMOSA architecture is shown in Figure 3, and the model content at each combination of views and life cycle in Table 1.
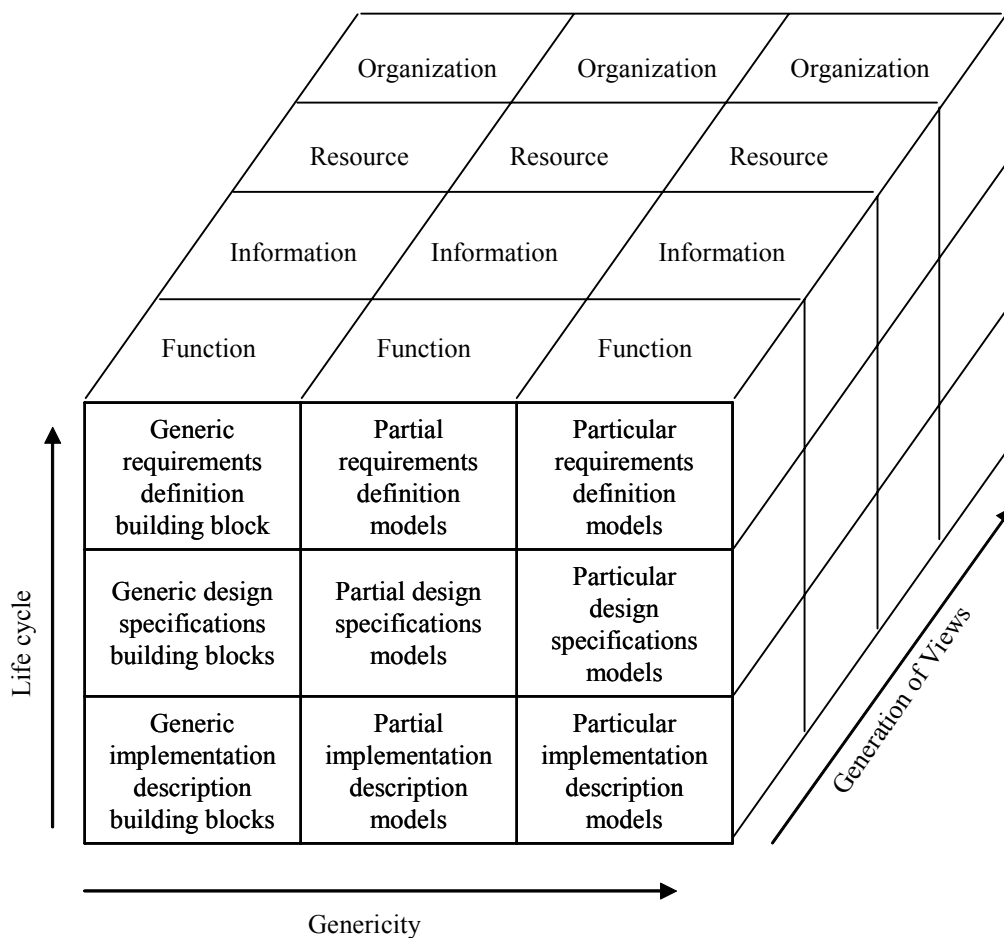
Figure 3: CIMOSA Architecture (Bernus & Nemes, 1997)

Table 1: CIMOSA Views and Life Cycle (Li & Williams, 1994)

Views	Life Cycle		
	Requirements definition	Design specification	Implementation description
Function	Domain processes and business processes; events	Specified functional operations	Implemented functional operations
Information	Enterprise objects and relationships; information elements (integrity rules)	External scheme; conceptual scheme (integrity constraint, database transactions)	Implemented external scheme and conceptual scheme (logical data scheme and physical data scheme).
Resources	Capabilities	Specified capabilities, resources, and resource units	Implemented capabilities, resources, and resource units
Organization	Responsibility; authority	Organization units and cells	Implemented organization units and cells

The goal of the proponents of CIMOSA was to present a new way to engineer and maintain enterprise systems and to introduce a new discipline: enterprise engineering and integration (Kosanke & Zelm, 1999; Zelm & Kosanke, 2001). The proponents of CIMOSA developed two different methodologies: one for the expert modeler who engineers the enterprise and develops enterprise models, and another one for the business user who uses models for supporting his or her work and evaluating operational alternatives. The methodology for the expert encompasses the entire life cycle, and it includes all enterprise levels and views, such as domain establishment, operational and behavioral analysis of business processes, information, resources, organization, and a consistency check among all views. The methodology for the business user works with an existing model to modify it according to decision needs; a business user does not participate in designing or implementing models.

The ultimate goal of CIMOSA is to provide a model-driven approach for operations support, monitoring and control. This in turn requires the support of an Integrating Infrastructure and an IT platform to execute CIMOSA process models in heterogeneous manufacturing and IT

environments (Kosanke, 1995). CIMOSA has been applied in re-engineering processes in European industries (Zwegers & Gransier, 1995).

3.3.3 Purdue Enterprise Reference Architecture

The Purdue Enterprise Reference Architecture (PERA) provides a framework (or reference architecture) and an enterprise integration (EI) process. PERA mentions the importance of strategic aspects. For PERA, enterprise integration is a small part of enterprise engineering, and the most important goal of enterprise engineering is to engineer the total enterprise throughout its life cycle (Li & Williams, 1994).

PERA and its methodology were developed at Purdue University starting in 1989 as part of the work on the Industry-Purdue University Consortium for CIM. PERA is based on the Purdue Reference Model for CIM and on earlier work of the Purdue Laboratory for Applied Industrial Control started in the mid-1970s. PERA focuses on the life cycle concept, or Enterprise Engineering process (Bernus & Nemes, 1996). PERA adheres to the following concepts of systems engineering in enterprise integration: applicable to any type of enterprise; the enterprise must have a mission; separation of mission fulfillment and control functions; information and physical processes are performed in a network of tasks. PERA classifies only two types of processes: those related to production and to services that fulfill the enterprise mission, and those related to the control of the mission and taking care of achieving the mission in an optimal manner (Williams & Li, 1998).

The PERA life cycle consist of nine phases (Bernus & Nemes, 1996): (1) identification, (2) concept, (3) definition, (4) functional design, (5) detailed design, (6) construction and

installation, (7) operation and maintenance, (8) renovation or disposal, and (9) enterprise dissolution. A contribution of PERA is the decomposition of these phases considering the types of resources involved, which leads to the analysis of the following twenty-eight areas of the enterprise (Williams, 1998):

- Identification: (1) Identification of enterprise business entity.
- Concept: (2) identification of business entity, its mission, vision, values, operational philosophies, and mandates.
- Definition: (3) production policies regarding customer, operational (product, service, manufacturing), goals, and objectives; (4) Information policies: related operational policies, goals, and objectives; (5) production requirements, to be fulfilled by the customer related policies; (6) information requirements, to be fulfilled by the information related policies; (7) Production functions: sets of tasks, functions, modules, and macrofunction modules required to carry out the customer related requirements of the enterprise mission; (8) information functions: sets of tasks, functions, modules, and macrofunction modules required to carry out the information related requirements of the enterprise mission; (9) process flow diagrams showing the connectivity of tasks, functions, modules, and macrofunctions of the manufacturing or customer product and service processes involved; (10) process flow diagrams showing the connectivity of tasks, functions, modules, and macrofunctions of the information or mission support activities.

Starting with Phase 5 (functional design) PERA divides the analysis in three subsystems: the manufacturing subsystem, the human and organizational subsystem, and the information and control subsystem.

- Functional design: (11) functional design of the manufacturing or customer product and service equipment architecture; (12) functional design of the human and organizational architecture. Establish the extent of the involvement of humans and automation; (13) functional design of the information systems architecture (entity-relationship diagrams).

- Detailed design: (14) detailed production equipment: design of components, processes, and equipment of the manufacturing or customer product and service equipment architecture; (15) detailed design of the task assignments, skill development, and training plan; (16) detailed design of hardware and software of the information system architecture.

- Construction/Implementation: (17) construction, checkout, and commissioning of the equipment and processes of the manufacturing equipment architecture; (18) implementation of organizational development training courses and on-line skill practice for the human and organizational architecture: staffing, training, checkout plant procedures; (19) construction, assembly, test, checkout, and commissioning of the equipment and software of the information systems architecture.

- Operations and Maintenance: (20) Production: continued improvement of process and equipment operating conditions of the manufacturing or customer product and service equipment architecture; (21) operation and maintenance: ongoing training, performance improvement, continued organizational development of skills and human relations in the human and organizational architecture; (22) operation of the information and control system of the information systems architecture including its continued improvement: maintenance, debug and upgrade.

- Renovation: (23) Review of mission for enterprise. Planning for revamping and redesign of customer product and service production equipment. (24) Review of mission of

enterprise. Planning for revamping and redesign of organizational architecture as mission changes. Retraining of personnel as new tasks and new skills require. (25) Review of mission for enterprise. Planning for revamping and redesign of information systems. Preservation and transfer of system information as needed.

- Dissolution: (26) disposal of physical equipment in ways which optimize economics without major injury to environment if the decision is made to discard customer product and service plant and equipment; (27) take necessary legal steps to dissolve charter of former enterprise; reassignment of any remaining personnel; (28) disposal of information systems and control equipment in ways that are benign to the environment while pursuing best-related economics.

PERA has developed specific interfaces for the place of humans in the enterprise. GERAM was developed from CIMOSA, GIM, and PERA, so GERAM also considers such interfaces but the presentation of GERAM is different due to the influence of CIMOSA and GIM. The authors of PERA developed an "Implementation Procedures Manual" for laying out requirements for the integration of the enterprise system; this manual guides the formulation of a Master Plan, the initial step in any CIM or any systems engineering project (Williams et al., 1996; Williams et al., 1998; Williams, 1999). See the PERA enterprise life-cycle model in Figure 4.
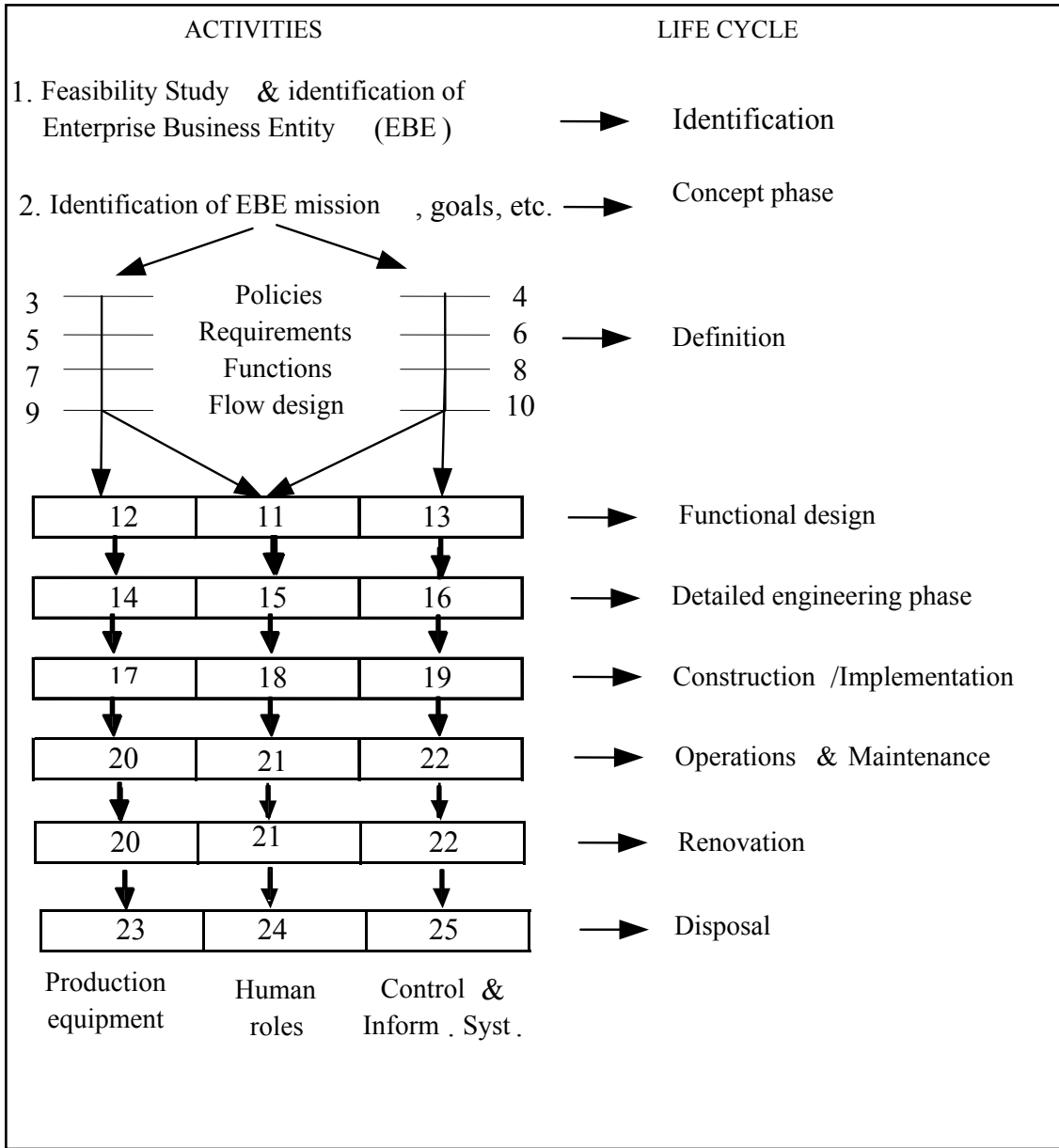
Figure 4: PERA Enterprise Life-Cycle Model (Williams et al., 1996)

3.3.4   Generalized Enterprise Reference Architecture and Methodology

The International Federation of Automatic Control and the International Federation for Information Processing (IFAC/IFIP) Task Force on Architectures for Enterprise Integration defined the Generalized Enterprise Reference Architecture and Methodology (GERAM).

GERAM does not impose tools or methods, but defines criteria for any tool or method to be used in enterprise engineering and integration. Kosanke et al. (1998) stated that GERAM is meant to unify existing architectures rather than replace them. GERAM is a framework for comparing and checking completeness of architectures and methodologies in the enterprise integration field. It does not have its own constructs and methodology, so it can not be directly applied in an enterprise (Ortiz et al., 1999).

Details about the components of GERAM have been included as an informative appendix in an international standard for requirements for enterprise reference architectures and methodologies. According to this standard, the components of GERAM are (ISO, 1999b; Bernus & Nemes, 1997):

- Generic Enterprise Reference Architecture (GERA).

- Enterprise Engineering Methodologies (EMLs).

- Generic Enterprise Modeling Concepts (GEMCs).

- Partial Enterprise Models (PEMs).

- Particular Enterprise Models (EMs).

- Enterprise Engineering Tools (EETs).

- Enterprise Modules (EMOs).

- Enterprise Operational Systems (EOSs).

The *Generic Enterprise Reference Architecture (GERA)* identifies concepts for enterprise engineering and integration. GERA resulted from the evaluation and integration of three major reference architectures for CIM: CIMOSA, GRAI-GIM, and PERA. GERA consists of a life cycle (identification, concept, requirements, preliminary and detailed design,

implementation or build, operation, and decommission), four modeling views (function, information, resources, organization/decisional), and three levels of genericity (generic, partial, and particular). It has two types of activities, customer oriented and control oriented; and two main agents responsible for performing processes, machines or humans (Vernadat, 1996; Bernus & Nemes, 1996; Kosanke et al., 1999).

Enterprise Engineering Methodologies (EMLs) describe generic descriptions of the processes for enterprise engineering and integration. *Generic Enterprise Modeling Concepts (GEMCs)* define generic concepts for enterprise modeling (i.e. semantics). Generic enterprise models capture concepts common to all enterprises (Kosanke & Nell, 1999b). *Partial Enterprise Models* (PEMs) capture common characteristics in an industrial sector, or across several industrial sectors. *Particular Enterprise Models* (EMs) describe a specific enterprise. *Enterprise Engineering Tools* (EETs) support methodologies, languages, analysis, design, and use of enterprise models. *Enterprise Modules* (EMOs), or generic enterprise modules are standard implementations of components that can be used to implement an enterprise, such as human or manufacturing resources, and IT. *Enterprise Operational Systems* (EOSs) support the operation of a particular enterprise.

One of the features of GERAM, inherited from PERA, is the concept of life cycle applied to any enterprise entity. GERAM and PERA life cycles include two phases that relates strategy with the engineering of an enterprise. These phases are enterprise entity identification and enterprise entity concept. The identification phase may be considered an entrepreneurial exercise because it is concerned with setting up the nature of an enterprise, its boundaries, internal and external relationships, and satisfying a market need. The concept phase is

47

related to the definition of mission, vision, values, strategies, objectives, policies, and operational concepts, which are strategy-oriented business activities (ISO, 1999). See the GERA architecture in Figure 5.
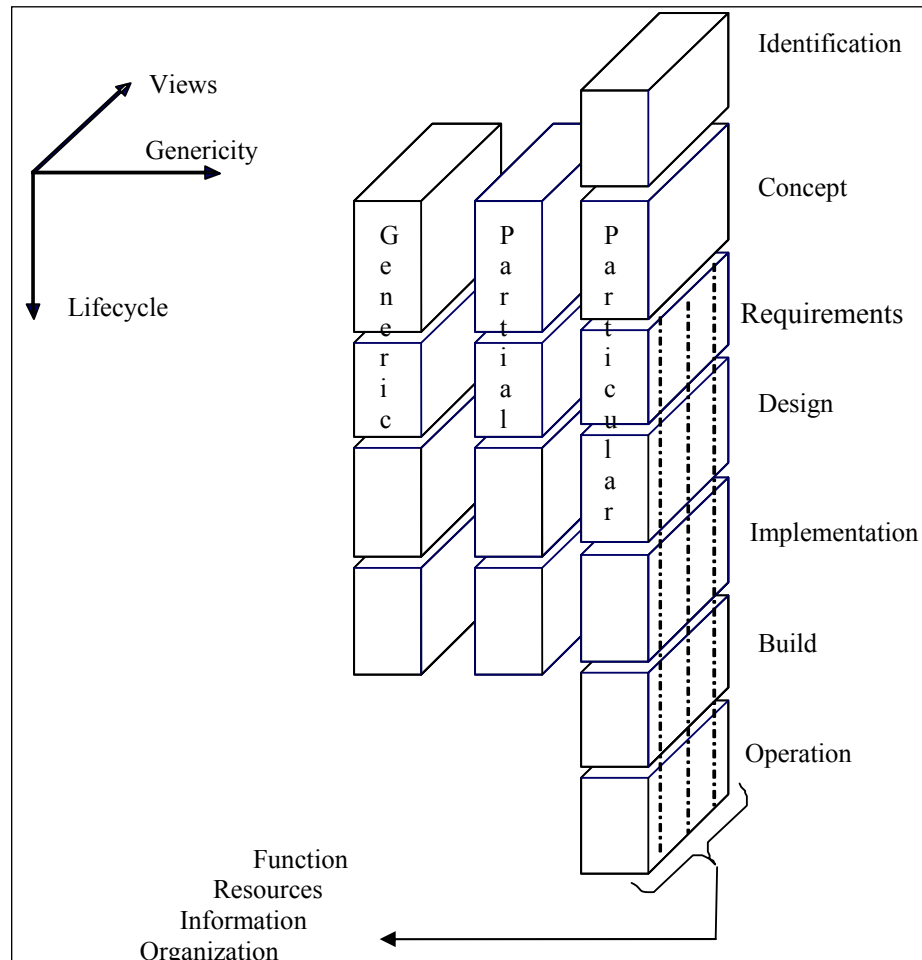
Figure 5: GERA Architecture (Vernadat, 1996)

A contribution of GERAM is that the proposed life cycle-based methodology is applicable to (Bernus & Nemes, 1996; 1997):

- The life cycle of the products produced by an enterprise.

- The enterprise life cycle. It could be a product-producing enterprise, a short-lived process, or a project that ends when a goal is achieved.

- The life cycle of an enterprise engineering and integration process.

- Strategic enterprise management process life cycle, whereby the need for change or creation of a new enterprise is identified and decisions are made to undertake an enterprise engineering and integration process.

Similar to CIMOSA, GERAM does not impose a defined set of views, allowing for representation of all the relevant aspects of an enterprise. GERAM intends to relate other change methods, such as BPR, TQM, and concurrent engineering, and improve communication among different disciplines contributing to enterprise integration (Bernus & Nemes, 1996).

3.3.5   Zachman's Framework

Literature on the Zachman Framework was published as early as 1987. The Zachman Framework for Enterprise Architecture classifies enterprise models by two basic aspects: the intended audience and the content of the model. The former is similar to *life cycle*, whereas the latter is similar to the *views* in other enterprise architectures. Five intended audiences together with six content's descriptions form the framework. From the audience perspective this framework includes (Zachman, 2003):

- *Planner*: establishes the system scope, boundaries, order of magnitude, relevant constituents, and provides a contextual perspective.

- *Owner*: establishes a business model, how the final product is going to be used by its users, and provides a conceptual perspective.

- *Designer*: establishes a logical model of the systems, an engineering view that discriminates between what is desirable and what is technically or physically possible.

- *Builder*: establishes a technology model, produces the end product under technological constraints. It has a physical perspective.

- *Sub-contractors*: provides detailed representation and product specifications, including data definition, program (language statement), network architecture, security architecture, timing definition, and rule specifications.

From the model content perspective this Zachman's framework describes a system in terms of six contents (Zachman, 2003):

- *Data*: the important objects to store data about, data models and relationships.

- *Function*: functional specifications; business processes that perform the transformation (input/outputs). The data and function contents are analogous to the information and function views in CIMOSA and GIM.

- *Network*: spatial description; components localization related to one another. The logistics and network models for enterprises.

- *People*: operating instructions, people and workflow models for enterprises, focusing on who does the work.

- *Time*: focusing on when events happen (timing) and life cycles.

- *Motivation*: the end, strategies for enterprises, similar to a control view.

The Zachman's framework (Table 2) perspectives do not match exactly with the life cycle of other enterprise architectures such as CIMOSA, GERAM or PERA. The perspective named "scope" in Zachman's framework is similar to the identification and concept phases in

GERAM and PERA, while the remaining perspectives are analogous to Requirements, Design, and Implementation life cycle phases in GERAM and PERA. Two distinctions with other architectures are that the Zachman's framework focuses on what PERA calls the "information systems architecture", and it specifically includes a perspective called "motivation", dedicated to goals and strategy (Zachman, 2003).

Table 2: Zachman Framework for Enterprise Architecture (Zachman, 2003)

PERSPECTIVE / INTENDED AUDIENCE	MODEL (CONTENT)					
	Data (What)	Function (How)	Network (Where)	People (Who)	Time (When)	Motivation (Why)
Scope (contextual) / Planner	Things important to the business	Processes performed	Location to operate	Major organization units)	Events/ cycles	Goals/ strategies
Business model (conceptual) / Owner	Semantic model (business entity & relationships)	Business process model; input/output resources	Business logistic system (location-linkage)	Workflow model (organization unit - work product)	Master schedule (event-cycle)	Business plan (objective-strategy)
System model (logical) / Designer	Logical data model (data entity-relationship)	Application architecture (application function-user views)	Distributed system architecture	Human interface architecture (role - deliverable)	Processing structure (event-processing cycle	Business rule model
Technology model (physical) / Builder	Physical data model (table-keys)	System design (Computer function - data elements)	Technology architecture hardware & software / line specifications	Presentation Architecture (User - screen format)	Control structure	Rule design (condition - action)
Detailed representation (out-of-context) / Subcontractor	Data definition (Field-address)	Program (language statement-control block)	Network Architect. (Address-protocol)	Security Architecture (Identity - Job)	Timing definition	Rule specification (sub-condition - step)

3.3.6   Architecture of Integrated Information System

According to its author, ARIS is suitable for fully describing standard software solutions, integrating methods for modeling information systems, developing methods for describing

and managing business processes, and providing a framework for describing the assembly of software components. Its proponent have asserted that it is ideal for configuring workflow systems (Scheer, 1998; 1999).

ARIS is similar to CIMOSA and GERAM in terms of modeling views and modeling levels (Vernadat, 2001). ARIS calls its life cycle a "*phase model*", starting with business descriptions and ending with objects for information and communications technologies. Phase 1 establishes the initial strategic situation. Phase 2 involves requirements definition. Phase 3 entails design specification. Phase 4 is the implementation description. ARIS links enterprise strategy with information management. The life cycle encompassed in the four phases is very similar to that of CIMOSA. According to its author, an advantage of ARIS over CIMOSA is that the control view and the description of interfaces among all the views allow the reassembling of the entire context (Scheer, 1998; 1999).

In contrast with CIMOSA, the genericity – a word coined in the field of enterprise engineering to denote genericness – of the models is not addressed directly by ARIS; rather it is done in the granularity of the information model.

ARIS provides modeling approaches and meta-models for five individual views, namely function, organization, data, output, and control:

- *Function*: it models the processes transforming inputs into outputs. It considers the enterprise goals given that processes support goals, and goals control processes.
- *Organization*: it models the hierarchical structure that groups entities responsible for the execution of work.

- *Data*: it models the data, messages, and their processing environment.

- *Output*: this view describes the physical and nonphysical flows.

- *Control*: it models relationships among all the other views (function-organization, function-data, function-output, organization-data, organization-output, and data-output), and among business processes. See ARIS architecture and its phases in Figure 6.

Each view in ARIS provides a specific modeling and representation capability. The function, organization, data, and output views describe the system structure, whereas the control view describes the dynamic behavior of the business process flows (Scheer, 1998; 1999).
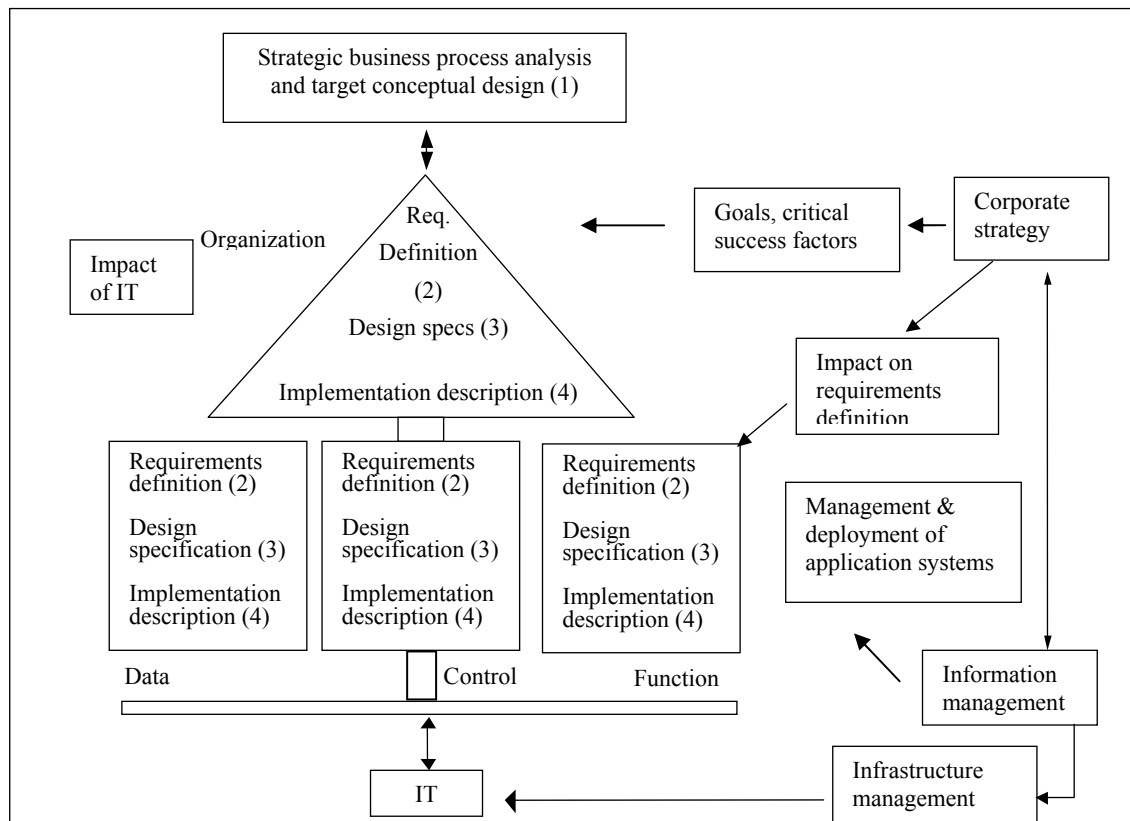
Figure 6: ARIS Architecture (ARIS House) (Scheer, 1998; 1999)

The ARIS architecture considers strategic business process analysis as an umbrella that covers further development. Strategy provides goals, critical success factors, and influences information management and the requirements definition of the future information system. ARIS developed event-driven process chains to represent business processes, a result used by SAP, the market leader in Enterprise Resource Planning (ERP) Systems (Appelrath & Ritter, 2000).

3.4 Enterprise Strategy

This section presents a review of strategy, strategic management, and elements that support the need to link strategy and ESE. According to Porter (1996), strategy is the creation of a unique and valuable position, involving a different set of activities, making trade-offs in competing, and creating fit among activities. Activities are the basic elements for creating competitive advantage; they deliver a unique mix of value to customers. Strategy is different from operational effectiveness; operational effectiveness is about achieving excellence in individual activities or functions. Similarly, Kaplan and Norton (2000) regard strategy as the unique and sustainable way in which organizations create value. These viewpoints of strategy have much in common with ESE, given that creating and combining activities to achieve synergy and create value streams are responsibilities of ESE.

3.4.1 Strategy

Strategy is a series of decisions and actions – not a single action – aimed at achieving the enterprise goals through activities that implement those decisions. Goals serve as the coordination mechanism among the decisions and actions throughout an enterprise.

Activities are the place where the organization's skills and resources must be aligned with the opportunities and threats in its environment (Coulter, 2002).

There are three levels of strategy:

- Corporate.

- Competitive.

- Operational strategy.

Kotler (1994) stated that *corporate strategy* aims at defining the company's mission, planning new businesses and setting business units, and assigning resources to these business units. The enterprise mission is the fundamental purpose of the enterprise, usually expressed in terms of products to make, markets to serve, and its role in the business environment. The vision establishes the future state of the enterprise in terms of competencies, capabilities, products, and markets (Martin, 1995). Corporate strategies proposed by David (1997) are: forward integration, backward integration, and market development. Other corporate strategies are merger, acquisition, takeover, stability, spin-off, and bankruptcy (Coulter, 2002). ESE is intrinsically linked to corporate strategy whereby corporate strategy focuses on the long-term, selecting the business which the company will pursue, and constraining the business processes that will support that corporate strategy and the configuration of those business processes.

Generic *competitive strategies* are cost leadership, differentiation, and market segmentation (Porter, 1985). Porter (1985) offered other competitive strategies based on competitive position: variety-based positioning, needs-base positioning, and access-based positioning.

Mosey et al. (2003) described three schools of thought of competitive or business strategy. One of them emphasizes positioning. The generic competitive strategies, cost, differentiations, and market segmentation are representative of the Positioning school. Within this view, a firm must have a unique strategic position in the market to achieve competitive advantage. The other two schools are based on strategy formulation and resources. The former supports the strategy formulation process and argues that the existence of a process to formulate strategy is fundamental for competing. The latter is based on resources, proposing that a firm must have superior, inimitable resources spanning the enterprise functions in order to compete effectively. The resource-based view of strategy conceptualizes an enterprise as sets of resources and capabilities and superior performance is achieved through the ability to exploit and deploy resources (Kalpic et al., 2003).

Kalpic et al (2003) offered different types of business strategies including product differentiation strategies, product diversification strategies, and generic market strategies based on location, maturity, and relationship with competitors.

Operational strategies are formulated and implemented at each main functional area of an enterprise (e.g. marketing, finance, and manufacturing). Operational strategies must be integrated among themselves to support the overall company's objectives and create sustainable competitive advantage. Elements of manufacturing strategies are: production process (product or process focused), capacity, location, layout, integrated manufacturing, and inventory management systems (Coulter, 2002). Others propose operational strategies in a broader sense by involving more than one functional area in applying a certain operation strategy, e.g. low cost provider, high quality provider, stress customer service, rapid

introduction of new products, maintain reserve capacity, centralized or decentralized processing, stress mechanization, or prioritize employees stability (David, 1997). In the same broader operational context, Miller and Roth (1994) classified manufacturing companies according to three types of manufacturing strategies: "Marketeers", which seek to obtain market oriented capabilities as broad distribution, broad product lines, and responsiveness to volume changes; "Caretakers", which put low emphasis on developing strategic capabilities; and "Innovators", which emphasize changing designs and introducing new products quickly. The marketers were later named "Designers" and the innovators "Specialists" by Frohlich and Dixon (2001).

A extensive review on manufacturing strategies is presented by Dangayach and Deshmukh (2001) whereby different connotations are given to the strategic intent of manufacturing: a strategic weapon; a sequence of decisions over time that enables to achieve a desired manufacturing structure, infrastructure and capabilities; a set of coordinated objectives and action programs aimed at securing long-term sustainable competitive advantage; the driving force for continual improvements in competitive requirements/priorities; the choice of a firm's investment in processes and infrastructure that enable it to make and supply its products to chosen markets.

There are many variations regarding operational strategies (Miller and Roth, 1994). The common themes that permeate the literature is that manufacturing is growing in importance, as measured by competitive capabilities like quality, flexibility, delivery, and cost, and that operational strategies should be linked to business strategy. Due to the abovementioned, the

eventual implementation of corporate, business, and operational strategies will require the designing of integrated business processes, which is ESE's domain.

3.4.2   Strategic Management

David (1997) defined strategic management as the art and science of formulating, implementing, and evaluating cross-functional decisions that enable an organization to achieve its objectives. Strategic management is an interdisciplinary exercise, involving all the main enterprise functions, emphasizing interactions with the environment and among the enterprise functions (Coulter, 2002). Other authors have used the term *strategic planning* in lieu of strategic management (Kotler, 1994).   Kotler and Armstrong (2001) stated that strategic planning is the process of developing and maintaining a strategic fit between the organization's goals and capabilities and its changing environment. As long as maintaining this strategic fit involves changing the design of business processes, strategic planning is directly linked with ESE. David (1997) presented a detailed strategic planning process, or strategic management model, that includes:

- Mission: the purpose and scope of the enterprise's business in terms of products and markets.
- External (environmental) and internal audit.
- Long-term objectives, as market share, assets growth, sales growth, profits, and earnings per share.
- Strategy formulation.
- Policies: guides for decision-making.
- Annual objectives.

- Strategy implementation: includes resource allocation and performance evaluation (feedback and control).

Strategy formulation is the generation, evaluation, and selection of the means by which objectives will be achieved, thus it is an intellectual, analytical, and intuitive process focused on effectiveness. It considers all the available resources before taking any action but requires coordination among few individuals. *Strategy implementation* is about managing the enterprise during action; it focuses on efficiency and operational processes, and it requires leadership and coordination among many resources. Strategy implementation establishes policies, annual objectives, and allocates resources.

3.4.3   Links between Enterprise System Engineering and Strategy

The relationship between strategy and ESE has been documented by many authors in the enterprise engineering literature, particularly enterprise architectures. Martin (1995) discussed that the vision must be linked to the enterprise architecture to maximize long-term growth and effectiveness. Nell (2000) mentioned that integration investments are one element in the overall enterprise strategy to achieve enterprise goals. Vernadat (1996) goes further and directly assigned to enterprise integration the role of a strategy more than of a technology. The same author stated that enterprise integration consists on facilitating the material, information, decision and control flows throughout the organization. This is achieved by linking functions with information, resources, applications, and people. The aim is improving communication, cooperation, and coordination in the enterprise in order to have the enterprise behave as a whole and operate according to the enterprise strategy. The necessity for enterprise-wide integration can be explained by the need to keep business

operations aligned with strategy, share information, systems interoperation, estimate impact of decisions, and fast and effective response (Ortiz et al., 1999).

Ortiz et al. (1999) proposed an approach towards enterprise integration directly linked to strategic aspects: objectives of each business process, how each business process supports the enterprise strategy, and identification of parameters to measure results of business processes, all of which support the linking between conceptualization and operational effectiveness.

The strategy-ESE link is more readily seen in enterprise architectures. The enterprise architecture is considered a foundation for managing modern enterprises, a baseline to manage change, and it provides a mechanism for aligning the enterprise system. The enterprise architecture is one of the building blocks of an effective strategy (Whitman et al., 2001). Veasey (2001) stated that emerging enterprise architectures, such as CIMOSA, PERA, and GERAM,) attempt to provide coherence to strategy implementation by sharing a common model and language of the enterprise. CIMOSA, PERA, GERAM, ARIS, and Zachman's architecture consider strategic components, in particular:

- The CIMOSA methodology includes the identification of enterprise domain, relevant business objectives, outputs to be produced and constraints (Kosanke & Zelm, 1999).
- The first two phases of PERA, identification and concept phases, include the following strategic aspects: identification of the enterprise business entity, mission, vision, values, operational philosophies, and mandates (Williams, 1998).
- The life cycle of the ARIS architecture includes in its Phase 1 the establishment of the initial strategic situation (Scheer, 1998; 1999).

- The motivations for defining the scope and the business model perspectives in the Zachman's architecture include the goals, strategies, and objectives (Zachman, 2003).

- GERAM recognizes a life cycle for the process of strategic enterprise management where the need for change or for creation of a new enterprise is identified. GERAM proposes that part of the results of a strategic management process is an enterprise engineering project or projects (Bernus & Nemes, 1996; 1997).

The relationship between ESE and strategy can also be found in the strategy literature, as is illustrated below:

- Kotler (1994) stated that strategies are developed to satisfy key stakeholders, aiming at critical business processes improvement, which in turn requires the alignment of the enterprise resources and organization.

- Kaplan and Norton (1996), focusing on people, stated that alignment is having the objectives of the individual human resources and those of the different organizational units aligned with the company objectives and strategy.

- For Porter (1996), strategy involves creating fit among activities, or combining activities so that they reinforce one another.

- Kettinger and Teng (1998) argued that any process must be aligned with strategy, managerial aspects, people, structure, and IT.

- Smith and Reece (1999) defined fit as the degree to which operational elements match the business strategy. They argued that external fit has a significant positive and direct effect on business performance, and that the fit of the operational elements with the strategy is more important than a particular choice of strategy.

The review presented in this section reinforces the argument that ESE is responsible for designing, combining and communicating activities and therefore, integrating them and creating fit among them to comply with a strategy.

3.5   IDEF Methodology

The Integration Definition for Function Modeling (IDEF0) methodology is used to model the ESE process. The following is a summary of the IDEF0 Methodology based mainly on information available at the National Institute of Standards and Technology website (NIST, 1993a). During the 1970s, the United States Air Force Program for Integrated Computer Aided Manufacturing (ICAM) identified the need for better analysis and communication techniques for people involved in improving manufacturing productivity. As a result, the ICAM program developed a series of techniques known as the IDEF (ICAM Definition) techniques, which over the years have served as the foundation for and IDEF family of modeling methods. Members of the family include:

- IDEF0: Integration Definition for Function Modeling to produce functional models.
- IDEF1: Integration Definition to produce information models. Later, it was extended towards data models and renamed IDEF1X.
- IDEF2: Integration Definition to produce dynamic models, i.e. the time-varying behavioral characteristics of the modeled system or subject area.
- IDEF3: to develop process flow and object state description.
- IDEF4: to engage in object-oriented design.
- IDEF5: an ontology description capture method (KBSI, 2003).

The intended use of the IDEF0 standard is for enterprise modeling, it provides a consistent means for establishing interrelationships among input, output, mechanism, and control, in the modeling of an enterprise engineering process. IDEF0 is a modeling technique independent of Computer-Aided Software Engineering (CASE) methods and tools, but it can be used in conjunction with those methods and tools. Use of this standard permits the construction of models comprising system functions, functional relationships, and data that support systems integration (NIST, 1993a).

The National Institute of Standards and Technology (NIST) stated that IDEF0 is an engineering technique for performing and managing needs analysis, benefits analysis, requirements definition, functional analysis, systems design, maintenance, and baselines for continuous improvement. IDEF0 models capture functions and their interfaces, and reflect how system functions interrelate and operate just as the blueprint of a product indicates how the different pieces of a product fit together (NIST, 1993a). IDEF0 has the following characteristics: generic, rigorous and precise, concise, conceptual, and flexible.

Although the standard describes IDEF0 as a modeling language it is more a notation composed of graphical symbols and text. IDEF0 uses a top-down, or functional decomposition approach. The two basic constructs are a function box, a.k.a. ICOM box that represents activities, and arrows to connect activities (NIST, 1993a; Vernadat, 1996). Hence, an IDEF0 model is a hierarchical series of diagrams that gradually display increasing levels of detail of functions and their interfaces within the context of a system (see Figure 7).
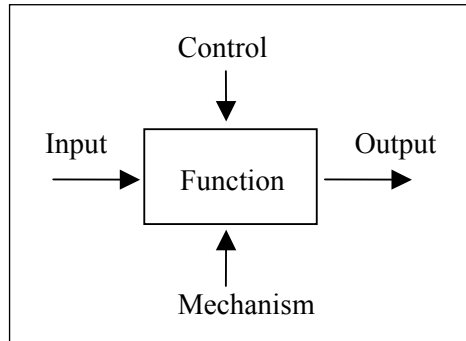
Figure 7: IDEF0 - ICOM Box (Vernadat, 1996)

Arrows represent inputs (I), controls (C), outputs (O), and mechanisms (M). *Inputs* are objects (physical or information) to be processed or transformed. *Controls* are used to activate, regulate, or synchronize the function (i.e. orders, constraints, schedules, management directives, and regulations). *Outputs* are objects processed or transformed by the function. When outputs are physical objects they can be used as inputs or mechanisms to another function. Similarly, when the output is information it can be used as input, mechanism, or control of another function. *Mechanisms* are physical resources or information needed to perform a function (Vernadat, 1996). Timing, sequencing, and decision logic are not included in an IDEF0 diagram.

The Integration Definition for Information Modeling (IDEF1X) methodology is used to model the relationships among the concepts used to define ESE. IDEF1X is a standard modeling technique for Federal Information Processing (FIPS) in the USA. According to the National Institute of Standards and Technology, IDEF1X is a modeling language with associated rules for developing information models, which represents the structure and semantics of information within a modeled system or subject area in an enterprise. IDEF1X produces graphical information models useful to support data management, integration of

information systems, and building computer databases. The primary objectives of this standard are to provide: a means for completely understanding and analyzing an organization's data resources; a means for representing and communicating the complexity of data; a technique for presenting an overall view of the data required to run an enterprise; a means for defining an application-independent view of data which can be validated by users and transformed into a physical database design (NIST, 1993b).

The building blocks of an IDEF1X model are entities, attributes of these entities, and relationships among entities. An entity is any single object (e.g. person, place, event, or concept) about which information is kept. Attributes are properties of an entity. A subset of attributes is chosen to identify each entity. Such subset is called a primary key. A relationship is a connection between two entities. Relationships have cardinality; which is a property stating how many instances of one entity may or must participate in a relationship with another entity (Bruce, 1992). Figure 8 shows an example of an IDEF1X diagram with a one-to-exactly one cardinality between two entities. The attribute in the top part of each entity is the primary key.
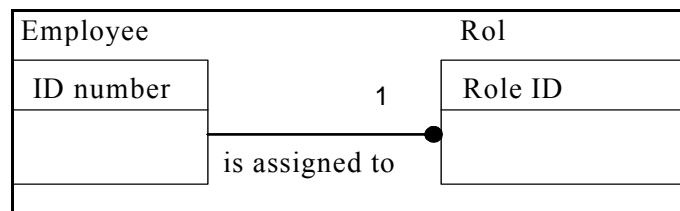
Figure 8: IDEF1X representation of two entities and a one-to-one or more cardinality

3.6 Queuing and Scheduling Notations

It is the intent of this research to develop a notation to identify areas within ESE. This notation will be similar to two existing ones: the Kendall's notation, used in queuing theory, and the notation used to describe scheduling problems. Queuing models model systems that present many variations. Existing queuing models capture some of these variations The Kendall's notation is a code of six ordered terms that identify the queueing system's variations. The terms represent the arrival pattern, the service pattern, the number of servers, the system capacity, the size of the population, and the queuing discipline. Letters, numbers, symbols, or acronyms are used to describe those six terms. A model identified by the terms M /E_2 /2 /5 /20 /FCFS represents a queuing system with Markovian (Poisson) arrivals, Erlang-2 service times, 2 servers, a system capacity of 5, a population of 20, and a queuing discipline of "first come, first served" (Ravindran, Phillips & Solberg, 1987).

Another notation comes from scheduling. Pinedo (2001) describes scheduling problems by a triplet α /β /γ:

- α (alpha) represents the machine environment. It is one single entry: 1 for single machine, P_m for m identical machines in parallel.

- β (beta) represents processing characteristics and constraints. It may have zero, one, or multiple entries: release dates, preemption, or precedence constraints.

- γ (gamma) represents the objective of the scheduling problem: minimize the completion time of the last job to leave the system, or the makespan.

3.7   Product Design and Development

Enterprises can be viewed as products, and as such they have to be designed, built, and put into operation (Bernus & Nemes, 1996). Consequently, product design theory may support the designing of enterprise systems. Focusing on the process of product design and development, Ulrich and Eppinger (2000) stated that the product development process traverses the following phases: planning, concept development, system-level design, detail design, testing and refinement, and production ramp-up.

Planning identifies market opportunities, market segments, product platforms, product architectures, new technologies, supply chain strategies, goals setting, production constraints, and general allocation of resources. *Concept development* identifies main customers, requirements, feasibility of product concepts, production feasibility and costs, and legal issues. *System level design* deals with the plan for the product family, alternative product architectures, major subsystems and interfaces, supplier identification and make-buy analysis. *Detail design* addresses market plan, parts geometry and tolerances, selection of materials, industrial design and documentation, production process for parts and assembling, tooling, and quality assurance. *Testing and refinement* includes developing market promotion and sales plan, field testing, reliability testing, obtaining regulatory approvals, implementing design changes, and refining fabrication, assembly, and quality assurance. *Production ramp-up* begins operation, places products with key customers, and evaluates production output.

For Suh (2001), design is the interplay between what the designer wishes to achieve and how to achieve it. Suh (2001) recognizes the product design spans over four domains: customer,

functional, physical, and process. There must be a progressive and ordered mapping between these domains starting with the customer and ending with the process domain. The mapping starts with the desired customer attributes which are translated into functional requirements. In turn, functional requirements are mapped into physical design parameters which are lastly mapped to process variables. Suh's (2001) methodology dictates that the designer zigzags between domains to find a solution that satisfies the functional requirements, design variables, and process variables.

Another product design approach useful in designing enterprise system is that of Quality Function Deployment (QFD). This is a method for ensuring quality throughout each stage of product development. QFD is concerned with the deployment of quality through the deployment of functions. Quality is understood as satisfying the customer, translating customer's demands into quality characteristics and these quality characteristics into design targets for the final product. Quality characteristics drive the manufacturing of each part and the process that manufacture them (Akao, 1990). The "Quality Chart" (Figure 9) summarizes the demanded qualities and it refines these quality demands until they can be measured. The QFD approach represents a chain of relationships (Kim & Moskowitz, 1997). In the columns of the Quality Chart design characteristics are broken-down into specific design elements, and the strength of the (qualitative) correlation between the demanded quality and the quality elements is indicated in the matrix (Akao, 1990). A triangle on the top of the matrix presents correlations of the design specifications among each other. Madu (2000) asserted that, as an analytical and hierarchical process that uses benchmarking, QFD is related to strategic planning.
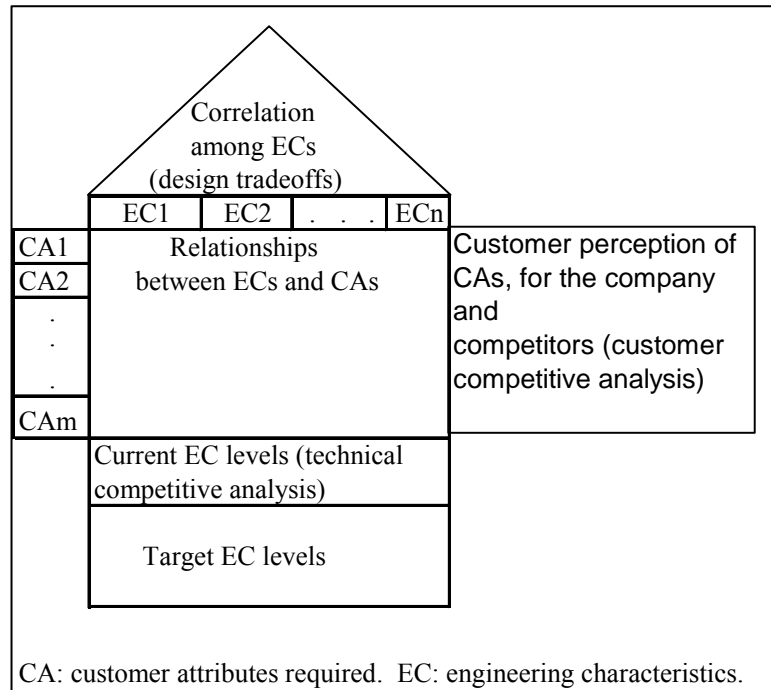
Figure 9: House of Quality Chart (Kim & Moskowitz, 1997)

3.8   Petri Nets

There has been ongoing interest in Petri nets. They are powerful in representation and analysis of dynamic systems that exhibit concurrency, parallelism, synchronization, non-determinism, and resource-sharing features (Vernadat, 1996). An enterprise system possesses all of these characteristics. Place-transition nets are the ones of interest in this research. The following review comes from works by Vogler (1992), Vernadat (1996), Jin (1999), and van der Aalst (2002).

A Petri net is a bipartite directed graph with three types of objects: places, transitions, and directed arcs. Places are represented by circles and transitions by bars or boxes. Arcs represent flow relations, which cannot connect places to places nor transitions to transitions.

Tokens, small black circles within the places are used to represent the dynamic behavior of a system. A Petri net is formally defined as 5-tuple:

PN = {P,T,I,O,M$_i$}, where

P = {p$_1$, p$_2$...p$_m$} is a finite set of places.

T = {t$_1$, t$_2$,....t$_n$ } is a finite set of transitions. $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$

I: (PxT) → N is an input incidence function that defines directed arcs from places to transitions, where N is a set of nonnegative integers.

O: = (TxP) → N is an output incidence function that defines directed arcs from transitions to places, and

M$_0$: P → N is the initial marking of the net, or the initial number of tokens in each place of the net.

The state of a system is defined by the number and distribution of tokens. A transition changes the distribution of tokens. When a transition fires it removes tokens from the input places connected to it, and it deposits tokens into the output places connected to it. The number of tokens to remove/deposit depends on the weights – capacities – of the directed arcs. An input place may be a precondition, input data, input signal, resource needed, condition, or buffer. A transition may be an event, computational step, signal processors, task, clause in logic, or processor. An output place may be a post-condition, output data, output signal, resource release, conclusion, or buffer. A transition without an input place is called a source transition. A transition without any output place is called a sink transition. A place and a transition are a self-loop if the place is both, the input place and the output place of the transition. A Petri net is pure if it has no self-loops. The set of all reachable markings

of a net from its initial marking can be represented by a tree called the reachability tree. See a Petri net in Figure 10.
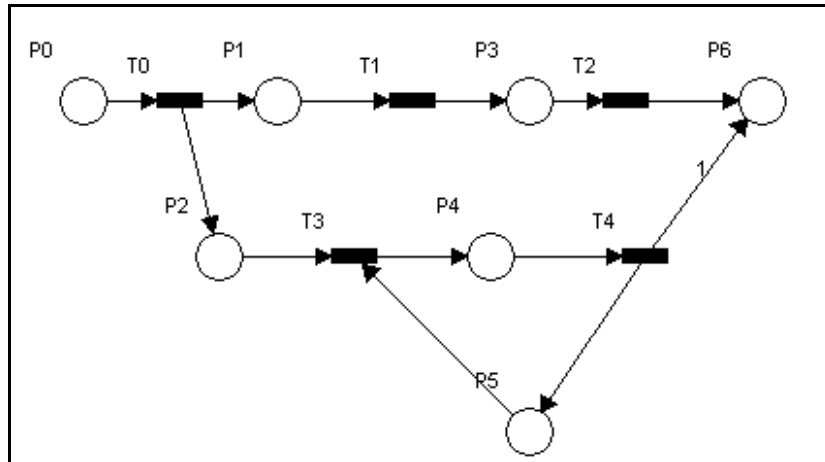
Figure 10: A Petri net (a place - transition net)

3.9   Literature Review Summary

This literature review has presented background information about how to develop definitions and provided the definition of enterprise system. It has also reviewed the works of many in the enterprise engineering area. Several conclusions can be drawn from these works:

- Most definitions of enterprise engineering suggest that: a) an enterprise is a system that evolves over time, b) it needs to work in an integrated manner, c) it has a life cycle, and it needs support to face rapid changes.

- Integration consists of linking resources that perform business processes. Enterprise integration is considered a subset of enterprise engineering. Most of the work on integration has been at the levels of physical integration and application integration. At business process level, the basic goals of integration are: improving overall system

efficiency, supporting coordination, supporting the achievement of the enterprise mission, and achieving higher responsiveness and effectiveness in the whole system.

- Enterprise architectures are the main source of frameworks and processes for engineering enterprise systems: a) the most known architectures are CIMOSA, PERA, GIM, and GERAM; b) they were created to support enterprise design, improvement, and business process integration; c) information systems architectures attempt to show in one single graphical representation all the enterprise components through the enterprise life cycle; d) enterprise architectures recognize the importance of strategy, however, different levels of strategy have not been incorporated into them.

This research used concepts form other areas such as operations research and information systems; thus, it reviews the IDEF0 methodology, queuing and scheduling notations, product design and development, and Petri Nets.

CHAPTER IV

ESE DEFINITION

This chapter introduces criteria to formulate definitions. Existing definitions of enterprise engineering are presented and evaluated against these criteria in order to gain insight and avoid pitfalls in formulating a definition for ESE. Note that this research does not redefine enterprise engineering; instead, it offers its own definition of ESE.

4.1 Specifications for Definitions

A basic tenet of this research is that an enterprise system and its components are viewed as products, for they have to be specified, designed, built, and put into operation (Bernus & Nemes, 1996); thus, the proposed definition has a product development orientation. The very nature of an enterprise system exudes complexity hence; the process to produce this product must also be complex. This complexity must be readily apparent in the proposed definition of ESE.

A theoretical definition is a statement of the essential nature of an object or concept. There are well-known specifications for defining objects (Beardsley, 1966; Copi, 1982; Copi & Burguess-Jackson, 1995; Chakrabarti, 1995). These specifications pertain to stating essential attributes, non-circularity, scope, affirmativeness, clarity, and simplicity. The proposed definition of ESE presented later is validated against these specifications:

- A definition must state essential attributes.
- A definition must be non-circular.

- A definition must have scope.

- A definition should have clarity.

- A definition should be affirmative.

- A definition should be simple.

Essential attributes are those related to the conventional connotation of the term, an intrinsic characteristic of it, its origin, its relationships to other objects or terms, or its uses. Non-essential attributes, called collateral characteristics, are linked to the essential attributes (Beardsley, 1966).

For a definition to be *non-circular*, the definiendum (i.e. the term being defined) cannot appear as part of the definiens (i.e. the terms explaining the definiendum). This specification rules out the use of synonyms and antonyms as part of the definiens. It also rules out the concatenation of definitions that at the end refer to themselves. Robinson (1968) further explained circularity as a flaw in analysis consisting in representing an object as a synthesis of elements one of which is itself.

The *scope* of a definition must be neither too broad nor too narrow. A broad definien denotes more objects than the definiendum intends to. Too broad a definition incorporates in the definien attributes that belong to other definienda (e.g. "an apple is a fruit"). A narrow or too exclusive definition denotes fewer objects than the definiendum is intended for. A narrow definition states in the definien an attribute that exists only in a subset of instances of the definiendum (e.g. "an apple is a red fruit"; yellow apples excluded in this definition).

A definition has *clarity* if it is literal and unambiguous, and it does not have obscure or metaphorical language (Beardsley, 1966). Ambiguity relates to a word having two or more distinct meanings in the same context (Copi, 1982; Copi & Burguess-Jackson, 1995).

Developing an *affirmative* definition is considered a preference more than a rule. Sometimes the complexity of a concept forces that it be defined in terms of what it is not. When possible, a definition explains what the definiendum means instead of what it does not mean.

Simplicity calls for a definition to be concise, yet complete. Analogous to Copi's (1982) statement that simpler hypotheses tend to be more accepted, Chakrabarti (1995) suggested that *simplicity* of a definition can be checked by a test of economy. That is:

- Economy of presentation: epistemically prior is preferred to epistemically posterior; which means that a term in the definien providing more knowledge is preferred, and an observable definien is preferred to unobservable one.

- Economy of relationship: using a term directly related to the definiendum is preferred over using a term indirectly related to the definiendum.

- Economy of constitution: a definition must not contain anything after the definiendum is correctly distinguished. A definition with fewer constituents is preferred, but it must have enough information for the purpose at hand (Beardsley, 1966).

4.2   Proposed Definition for ESE

Current understanding of the terms enterprise and system were presented in the literature review. This section adds a discussion of the term engineering, and it shows that the understanding of the three individual terms "enterprise", "systems", and "engineering" is

different from any combination of the terms: "enterprise system", "systems engineering", and "enterprise engineering". Afterwards, definitions of EE are shown and analyzed against the specifications for formulating definitions. The analysis and understanding of the strengths and limitations of existing enterprise engineering definitions facilitate a final formulation of a definition for ESE.

From the literature review, an *enterprise* is a system. A *system* is a set of interrelated parts working towards a common objective. A system is characterized for having a hierarchical structure, properties attributed to the whole not to the parts, and its control mechanism. A system where humans are involved has customers, performs some transformation, and it is subject to external constraints. *Engineering* has been defined as the systematic design and building of a process or an article from concept to a set of specifications that can be implemented. This systematic design and building uses science and mathematics (Jayachandra, 1994).

Based on the meanings of the terms enterprise, system, and engineering, relationships among them can be identified (see Figure 11). These relationships show that an enterprise has a mission and vision that guide the setting of strategies, goals, and objectives, which in turn guide and constrain the setting of business processes. Business processes deliver value to customers via the required products and services, and they deliver performance required by stakeholders. An enterprise is a system and as such has an owner, actors, structure, emergent properties as a whole, a control mechanism that measures performance and adjusts the system behavior, it pursues some objectives, performs some transformations, it is guided by a world view, and it is subject to environmental constraints (Checkland & Scholes, 1990).

76
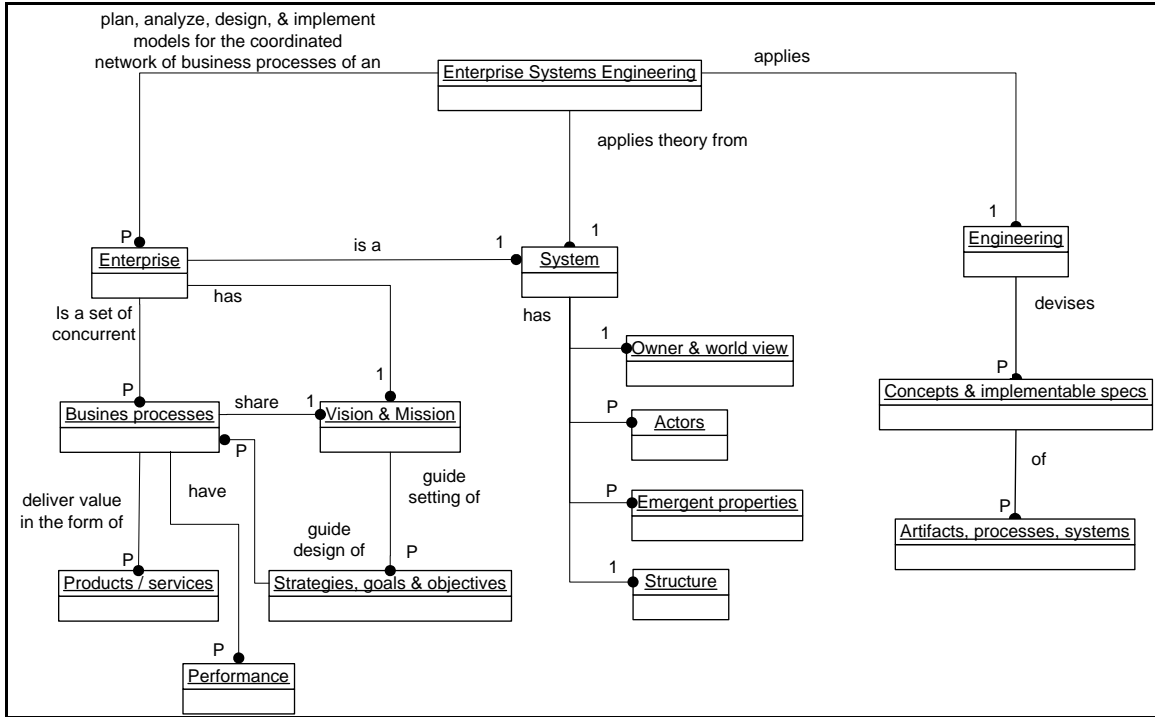
Figure 11: Conceptual Relationships among Enterprise, Systems, and Engineering

The pairing of any combination of the words enterprise, systems, and engineering, has different meaning than just the concatenation of individual meanings. An *enterprise system* can be a part of a business process, a whole business process, or a set of business processes. It can also be a whole company working independently, several companies working as part of a partnership (as in a supply chain), or a virtual or extended enterprise. An enterprise system can be viewed as a product that needs to be specified, designed, built, and put into operation. The same is true for the words "systems engineering". Systems engineering attempts to define system behavior and to design system structure so that emergent behavior can be predicted and controlled within desirable bounds (Thomé, 1993; ISO, 1999a).

Several definitions have been found for the words "enterprise engineering". The term "enterprise engineering" is the one most closely related to this research (Saenz & Chen, 2004). Existing definitions of enterprise engineering differ in scope, means, and/or focus. They have commonalities, all propose some kind of life cycle, and most of them stress the importance of business processes. Seven definitions of EE have been analyzed.

According to Vernadat (1996), Enterprise Engineering (EE) is the art of understanding, defining, specifying, analyzing, and implementing business processes for the entire life cycle so that the enterprise can achieve its objectives, be cost-effective, and be more competitive in its market environment. Kosanke et al. (1999) emphasized the communication among the main elements of an enterprise and stated that EE defines, structures, designs, and implements enterprise operations as communication networks of business processes that comprise all their related business knowledge, operational information, resources, and organization relations. Enterprise engineering is a life cycle oriented discipline

The International Organization for Standardization assigned a broad scope to EE and defined it as the discipline applied in carrying out any efforts to establish, modify, or reorganize any enterprise (ISO, 1999b). In a similar way to ISO, the International Society of Enterprise Engineering (ISEE) defines EE as the body of knowledge, principles, and disciplines related to the analysis, design, implementation and operation of all elements associated with an enterprise. For the ISEE, EE includes modeling, cost analysis, simulation, workflow analysis, bottleneck analysis, Total Quality Management (TQM), Just-in-time (JIT), change management, and value added analysis (ISEE, 2003).

In contrast with other authors, Martin (1995) viewed EE as an integrated set of change methods. Martin (1995) classified five change methods corresponding to different enterprise levels of change: (1) continually improving individual tasks (TQM); (2) reinvention of existing processes (procedure redesign); (3) reinvention of end-to-end business processes looking for significant gains in effectiveness through structural changes (value-stream reinvention); (4) reinvention of the fundamental and integral structure of the entire enterprise, or cultural aspects, including increment or reduction of business units; and (5) strategic visioning, where the entire context is validated or changed. Complementing these change methods are two infrastructure change processes: the organization and culture development, and IT development. This view of EE emphasizes changing and improving an existing enterprise and mentions the need for a new type of professional – the Enterprise Engineer – with knowledge of change methods, technology, and strategy, together with personal and cultural skills.

Presenting a similar perspective to Martin (1995), IFIC-IFAC (2003) has defined EE as the discipline that organizes the knowledge, tools and methods needed to identify the need for change in enterprises, make the necessary design or redesign, carry out that change in a professional manner, and continually maintain an integrated state of the enterprise.

For Presley and Liles (1996) and Presley et al. (2001), EE involves the analysis, design, implementation, and operation of an enterprise. EE addresses the design and improvement of all elements associated with the total enterprise through the use of engineering and analysis methods and tools. Table 3 shows the seven definitions of EE in terms of their scope, the means suggested for addressing the field, and their focus.

Table 3: Comparison of Enterprise Engineering Definitions

Scope	Means	Focus
Understand, define, specify, analyze, and implement business processes for the entire life cycle (Vernadat, 1996)	Not specified, suggests enterprise modeling.	Business processes. Achieve objectives, be cost-effective and competitive.
Define, structure, design, and implement operations (Kosanke et al., 1999)	Business knowledge, operational information, resources, and organization relations.	Communication networks of business processes. Life cycle oriented discipline
Establish, modify or reorganize enterprises (ISO, 1999)	Any efforts.	Whole enterprise
Analysis, design, implementation and operation (ISEE, 2003)	Modeling, cost analysis, simulation, WF analysis, bottleneck analysis, TQM, JIT, change management & value added analysis.	All enterprise elements.
Engineer system for maximum benefit. Adapt to fast-changing demand (Martin, 1995)	TQM, redesign, reinvention (procedure, value-stream, whole enterprise) and infrastructure (organization + culture + IT).	An integrated set of change methods.
Organize knowledge, tools and methods needed to identify the need for change in enterprises (IFIP-IFAC, 2003)	Continually maintain an integrated state of the enterprise.	Make the necessary design or redesign, and carry out change in a professional manner
Analysis, design, implementation and operation of an enterprise (Presley & Liles, 1996; Presley et al., 2001)	Knowledge, principles, and practices.	Whole enterprise

The criteria for developing definitions were used to evaluate existing definitions of EE. For that purpose, values are assigned to each criterion as shown in Table 4.

Table 4: Values Assigned to Criteria for Formulating Definitions

Criteria	Assigned Values
State essential attributes	Yes: states essential attributes of the definiendum No: state collateral attributes of the definiendum
Non-Circularity	Yes: it is non-circular No: it is circular
Scope	Precise: denotes what it is intended Broad: includes more than intended, attributes belong to other definienda Narrow: includes a subset of the intended whole
Clarity	Yes: it is clear, literal, unambiguous, and non obscure language No: the opposite.
Affirmative	Yes: written in positive, state what it is No: states what it is not
Economy of presentation	Ok: enough information to convey and understand the concept Not ok: not enough information to convey/understand the concept
Economy of Relationship	Ok: definien is directly related to definiendum Indirect: definien is indirectly related to definiendum Not clear: not enough information to judge the intended relationship
Economy of Constitution	Ok: enough information is given to understand definiendum Indefinite: to little info is given to understand definiendum

An evaluation of these enterprise engineering definitions against the specifications for developing definitions reveals that Vernadat's (1996) and the IFIC-IFAC's (2003) definitions are the ones that best conform to the specifications. The one aspect in which these and all the other definitions fail short is in "scope". Five of the seven definitions are too broad; they include aspects related to operations management and other fields of study. A summary of the evaluation of existing definitions of EE against specifications for formulating definitions is presented is offered in Table 5.

Table 5: Evaluation of Existing Definitions of Enterprise Engineering

| Definition by | SPECIFICATIONS | | | | | | | |
| | State essential attributes | Non-circular | Scope | Clarity | Affirma-tive | Simplicity (economy of:) | | |
						Presenta-tion	Relation-ship	Consti-tution
Vernadat, 1996	Yes	Yes	Broad	Yes	Yes	Ok	Ok	Ok
Kosanke et al., 1999	Some essential, some collateral	Yes	Narrow, focused on integration	Yes	Yes	Ok	Indirect	Ok
ISO, 1999	No	Yes	Broad	Yes	Yes	Ok	Ok	Indefinite
ISEE, 2003	Yes	Yes	Broad	Yes	Yes	Ok	Not clear	Ok
Martin, 1995	Collateral	Yes	Narrow, focused on change methods	Yes	Yes	Ok	Ok	Ok
IFIC/IFAP, 2003	Yes	Yes	Broad	Yes	Yes	Ok	Ok	Ok
Liles, 1995, 1996	Yes	Yes	Broad	Yes	Yes	Ok	Not clear	Ok

The ISO (1999b) definition fails to state *essential attributes*. Kosanke et al. (1999) provide some essential attributes and some collateral attributes, whereas Martin's (1995) provides only collateral attributes. All definitions satisfy the *non-circular* specification. All definitions satisfy the specification of *clarity* and *affirmative*, and to some degree that of *simplicity*. Kosanke's et al. (1999), ISEE's (2003), and Presley and Liles' (1996) fail the economy of relationship, whereas ISO's (1999b) fails economy of constitution.

From the literature review and the previous analysis on definitions the following lessons have been learned:

- Emphasis has been in the business process side of the enterprise system.

- An enterprise system is made up of a coordinated network of enterprise elements, such as work, resources, information, and decision.

- The coordinated network of enterprise elements must be engineered throughout a life cycle, determines how efficiently and effectively the organization transforms its inputs into outputs, delivers value to customers, and is a function of the enterprise capacities and capabilities (Coulter, 2002).

- The interdependencies among the network of enterprise elements define system behavior. This network must be aligned with strategy and satisfy customer and stakeholder requirements, and thus achieve certain performance (Malone & Crowston, 1994).

- Furthermore, the engineering of an enterprise system is analogous to the engineering of a product. According to product design theory the development of a product has three different but complementary outcomes: the blueprints of the product itself and its components; the process plan to manufacture the product; and the product's assembly plan (Ulrich & Eppinger, 2000; Suh, 2001).

Based on these lessons and considering that an enterprise system is made up of a network of interrelated enterprise elements, this study defines ESE as

"*an engineering discipline that develops and applies systems theory and engineering techniques to specification, analysis, design, and implementation of an enterprise for its life cycle*.

Similar to other engineering discipline, ESE designs artifacts (i.e., enterprise systems) that meet the customer's need. To achieve its defined purpose, ESE develops and applies systems engineering tools and techniques for planning, specifying, modeling, analyzing,

designing, and implementing enterprise systems. Moreover, ESE aims at building a scientific foundation for study of the integrative and collaborative nature of enterprise behavior in the global economy.

4.3   Validation of the ESE Definition

In general, this research has focused on the fundamental descriptive and qualitative side of theory building, not on hypothesis testing (Beardsley, 1966). For validation purposes of the proposed definition, classification, scheme, and ESE process, a deductive approach to science has been used; the analysis of the available theory backed the outcomes of this research (Dubin, 1969). Validation of the proposed ESE definition was done in two fronts: 1) adherence to a specific technique for defining, and 2) compliance with scientific criteria for formulating definitions.

Using the *synthesis* technique, a concept can be defined by specifying its place in a larger system of concepts or expressing it in terms of other primitive concepts (Robinson, 1968). The ESE definition was synthesized by relating it to its three primitives (enterprise, systems, and engineering) and by relating it to a well established and accepted theory for developing products.

The proposed ESE definition complies with the six criteria for formulating scientific definitions. As opposed to all the existing definitions of EE the proposed definition focuses on two *essential attributes:* 1) it focuses on developing and applying systems theory and engineering techniques; 2) it states that the interest is the resulting whole that creates value, the enterprise. The evaluated definitions of EE contributed by highlighting a particular side

84

of the problem, but they tend to focus on the techniques (ISEE, 2003), remain generally broad (Presley & Liles, 1996; ISO, 1999b; Presley et al., 2001), stress the applications integration (Kosanke et al., 1999) or the achieving of change (Martin, 1995). The proposed definition has its origin in product development theory and all the mentioned collateral characteristics are related to it.

The proposed definition is *non circular*. The definiendum does not appears as part of the definiens. Synonyms are not used either. There is no concatenation of meanings that refer to themselves, and the synthesis of the proposed definition excluded elements that belong to the definiens. Thus, the proposed definition is non circular.

The main criticism for existing EE definitions is rooted in their *scope*. Two out of seven are considered narrow and five out of seven are considered broad (Table 5). This does not mean that they are incorrect, but it is argued that broad definitions do not give uniqueness to the field because they connote more than their definiendum intends to. Contrasting to this, the narrow definitions leave the feeling of excluding crucial aspects of ESE while at the same time specializing in a certain aspects that invade the realm of other engineering fields. The proposed definition has a precise scope: "*specification, analysis, design, and implementation of an enterprise for its life cycle.*"

The proposed definition has *clarity*. All the terms in the definition are expressed in clear, literal, unambiguous, and non obscure language. To further guarantee adherence to this criterion, key terms as design, enterprise elements, and value have been assigned a distinct and accepted meaning in this research to avoid ambiguity.

The proposed ESE definition as expressed is *affirmative* and direct; it is not expressed in negative terms. To test that the proposed definition is *simple* enough without being indefinite, three tests were checked: economy of presentation, economy of relationship, and economy of constitution (Chakrabarti, 1995). Regarding presentation, the definien provides enough information to convey and understand the concept of ESE. No attributes of the enterprise, the system it represents, the engineering process, or the possible methodologies to use are given because this is not a denotative definition. In some definitions, it is not clear when the definition ends nor if the enumeration of attributes is part of it, as in Martin (1995) and ISEE (2003). Regarding economy of relationships, all the terms used are directly related to the definiendum, that is, to the constituent terms enterprise, systems, and engineering, at the same time giving a new and clear meaning to the ordered set of terms. Regarding economy of constitution, the definien contains nothing beyond necessary to explain the meaning of the definiendum.

In short, the proposed definition of ESE states essential attributes, is non circular, has a definite scope, and is clear, affirmative, and simple. Therefore, it is a valid definition.

CHAPTER V

ESE CLASSIFICATION SCHEME

This chapter presents criteria for developing classifications, offers a specific classification scheme for ESE describing its components along with a notation, and presents its validation. A classification is instrumental in the discovery of new knowledge (Beardsley, 1966). Hempel (1965) stated that a classification in any domain of investigation may be considered a special type of scientific concept formation. In general, a classification is a systematic arrangement of objects into groups, categories, or classes (Merriam-Webster, 2004). The classification of any object is based on comparisons with established criteria, which examines similarities, differences, or analogies. The exploration of relationships among classes may result in a new classification of objects (Beardsley, 1966).

Classification schemes are part of logical analysis (Patton, 2002). According to Copi (1982), a classification is generally most important in the early stages of a science field. In the emerging field of enterprise systems engineering, a classification scheme is more of a necessity than merely a different approach. To date, there is no formal classification scheme for ESE, instead, graphical representations, known as enterprise reference architectures, have been used to guide the analysis, design, and implementation of enterprise systems. The proposed classification scheme goes beyond the limitations imposed by three dimensional graphical representations because it uses a tabular form and at this point has four dimensions.

5.1   Specifications for Classification Schemes

The specifications for a classification scheme have been defined using several sources including enterprise engineering and product design theory. Product design principles, concepts, and approaches are of general application, and they have been used to develop the proposed classification scheme for ESE. In particular, the principle that the beginning of every product development is the specifications, which represent the customer requirements (Ulrich & Eppinger, 2000; Suh, 2001), has been used and complemented with concepts developed by Copi (1982), and the work of philosophers Hempel (1965), Breadsley (1966), and Gay and Airasian (2000), to formulate and develop the proposed classification scheme.

One specification has been borrowed from enterprise engineering (Berio & Vernadat, 1999) and states that there must be a minimum content embedded in the classes of the classification scheme. This content must deal with flows, views, and *modeling levels*. *Flows* can be material, information, or decision (control). *Views* refer to different perspectives of the enterprise system such as function, information, resources, and organization. *Modeling levels* refers to development phases such as requirements, design, and implementation.

A classification divides a given set of k objects ($o_1, o_2,...,o_k$) into n classes ($C_1, C_2,...C_n$). The set of characteristics (H) that distinguish one class from another is called *the basis for division*. Breadsley (1966) stated that there must be only *one basis of division* to avoid confusion and the fallacy of cross-ranking, i.e. an object being classified in two different classes; in other words, if $o \in C_i \rightarrow o \notin C_j; i \neq j$. A significant classification has a basis for division made up of essential characteristics of the objects being classified. Other

characteristics, called collateral characteristics, depend on the essential characteristics. In this way, a classification describes objects from a point of view, an interest, or purpose (Hempel, 1965). For each class, there must be *membership criteria* specifying similarities or differences among the objects being studied (Hempel, 1965). Furthermore, *relationships among classes* must be stated. Two general relations are subordination (a class is located lower in the classification, inheriting attributes from its super class) and coordination (parallel classes, they have the same level in the classification) (Beardsley, 1966). A natural consequence is that relations form a hierarchy or network of relationships explicitly showing how each class relates to other classes. A classification must have a *hierarchy* of at least three levels (Gay & Airasian, 2000); otherwise, the classification is trivial and precludes analysis of the objects.

In summary, every classification scheme must satisfy the following specifications:

    (1) There must be a set of *n* classes, where $n \geq 2$.

    (2) There must be a clear and unique basis for division.

    (3) There must be fundamental distinctions among classes.

    (4) There must be criteria to establish membership in a class.

    (5) There must be relationships among classes.

    (6) There must be a hierarchy of at least three levels.

In addition, an ESE classification scheme must enable the description and analysis of flows, views, and life cycle.

5.2   Proposed Classification Scheme for ESE

Several sources were used for the development of a classification scheme for ESE, including existing theory on enterprise reference architectures and industrial cases. A comparative analysis of the main enterprise reference architectures was performed as a benchmark to pinpoint common themes and omissions. Industrial cases are valuable when refining theory, exposing complexities for further investigation, and helping to establish the limits of generalization (Verville & Halingten, 2002). The industrial cases were actual projects developed by the researcher (1990-2000) while working as project manager and consultant for the companies EuroConsult, S.A., Cooppers & Lybrand, Clapp & Mayne, and PriceWaterhouse Coopers, which required designing and redesigning business processes, and managing the development of information systems. The misfit between the needs of a practitioner of business process analysis/design and the frameworks to meet these needs provided the impetus for this research.

5.2.1   Comparative Analysis of Enterprise Reference Architectures

The three main enterprise reference architectures are GIM, CIMOSA, and PERA. These architectures were analyzed in regards to purpose, focus, life cycle, views, and abstraction levels.

Regarding *purpose*, GIM's purpose is mainly to support the design of CIM systems and other types of enterprises. CIMOSA strives to develop a model-driven approach to control business processes; ultimately, its goal is to produce formal, executable models that can be used for simulation and operation of the enterprise. PERA's purpose is to guide enterprise

integration, and it is the only architecture that is explicitly not targeted to computer science or information system users.

In regards to the *focus* of architectures, an enterprise system is made up of three subsystems: 1) a physical subsystem that delivers products and services; 2) a management subsystem that directs and controls; and 3) an information system that supports the other two. GIM focuses in the decision system, CIMOSA tends to focus on the representation of the enterprise system by using the information system and its own language, and PERA focuses in the physical system, that is, in its resources.

One common feature of GIM, CIMOSA, and PERA is that they have an explicit *life cycle*. In GIM, the life cycle phases are analysis, user oriented design, and technical oriented design. In CIMOSA, the life cycle phases are requirements definition, design specification, and implementation description. The last phase of GIM corresponds to design specification of CIMOSA. Among these three architectures PERA has the most extensive life cycle, with nine phases: (1) identification, (2) concept, (3) definition, (4) functional design, (5) detailed design, (6) construction and installation, (7) operation and maintenance, (8) renovation, and (9) disposal. The main differences among the life cycle of these enterprise reference architectures are that only PERA covers phases after operation (i.e. renovation or disposal, dissolution) and that GIM does not include construction and operation.

Views refer to models of a subset of the enterprise system. GIM has four views: functional, information, physical, and decision. CIMOSA also has four views: function, information, resources, and organization; however, it is open to include more views as needed. CIMOSA

includes the decision and physical views of GIM mainly in its organizational and resource views. PERA does not address views directly but deals with three subsystems: the manufacturing system that accomplishes the mission and produces services and products to costumers; the information system that supports the manufacturing system and management and control; and the human and organizational system. Each one of these three subsystems intertwines the resources and function views of CIMOSA and the decisions of GIM.

Considering the elements that the views highlight, *resources* and *information* are common elements to the three analyzed architectures. *Structure* is included in all the analyzed enterprise reference architectures. CIMOSA includes structure in its organization view, PERA has the organizational and human subsystem, whereas GIM has the decision view, which includes decision centers and decision levels. *Decision* is included as a separate view in GIM, as part of the function view in CIMOSA, and in the management and control view in PERA. The *work* to be done by the enterprise system is also a common element in the three architectures. Work is treated as part of the function view in GIM and CIMOSA, and intertwined in the three systems of PERA. *Flows* are included as part of the functional view in GIM and CIMOSA and in the manufacturing and information processes in PERA.

GIM has three *abstraction levels* labeled conceptual, structural, and implementable. CIMOSA does not have abstraction levels; instead, it has a genericity concept with three levels: generic, partial, and particular, for common models, applicable to any type of enterprise, for specific industries, and for specific enterprises respectively. PERA does not directly address abstraction levels; instead, it omits the identification, detailed design, construction, and operations life cycle phases from generic and partial models. See Table 6.

Table 6: Comparison of Enterprise Reference Architectures

	GIM	CIMOSA	PERA
Purpose	Design CIM systems and other types of enterprises	Develop a model-driven approach to control business processes; produce formal, executable models that can be used for simulation and operation of the enterprise	Guide enterprise integration. It is the only architecture that is explicitly not targeted to computer science or information system users
Focus	Decision subsystem	Representation of the system by using the information system and its own language	Physical subsystem and its resources
Life cycle	1) Analysis	1) Requirements: equivalent to analysis in GIM	1) Identification 2) Concept 3) Definition 1, 2 and 3 are partially included in the requirements phase of CIMOSA
	2) User oriented design 3) Technology oriented design	2) Design: equivalent to user oriented and technology oriented design in GIM	4) Functional design 5) Detailed design 4 and 5 are included in the design phase of CIMOSA
		3) Implementation: not included in GIM	6) Construction: equivalent to implementation in CIMOSA
			7) Operation and maintenance 8) Renovation 9) Disposal and legal dissolution 7, 8, and 9 are not included in CIMOSA
Views	1) Functional	1) Function	1) Manufacturing system; includes all the functions and resources for this system 2) Organizational & human; includes functions and resources for this subsystem 3) Information & control system; includes all the information; and functions and resources for this system
	2) Information	2) Information	
	3) Physical	3) Resources; equivalent to physical in GIM	
	4) Decision	4) Organization; includes decision in GIM	
Abstraction levels	Conceptual Structural Realizational	No abstraction levels, but three levels for genericity of models: generic, partial, and particular	Not specified

It is important to note that *abstraction* or *genericity* levels *per se* do not support enterprise modeling in terms of an additional dimension to model. Instead, the only action in regards to modeling is an arbitrary decision, made by the designer, to increase the level of detail, in the case of abstraction levels, or to customize the model for a particular industry, in the case of genericity levels. The absence of additional modeling activities in abstraction or genericity leads to the conclusion that the compared reference architectures actually have only two modeling dimensions for representing an enterprise system: views and life cycle.

In light of the comparative analysis, this research considers a generic abstraction level independent of industry types. For ESE, a generic framework is necessary with the understanding that abstraction level can be managed via the desired granularity of the models. The proposed classification scheme has four distinct classes and four subclasses within each class. Some of the classes used in the classification scheme are explicitly or implicitly included in at least one of the three main reference architectures; specially those pertaining from the identification up to the implementation activity. The classes and subclasses contain the objects and concepts needed as per the definition of ESE. The four classes and their subclasses and membership criteria are shown in Table 7.

A recurrent weakness of existing enterprise reference architectures is that they fail to incorporate an explicit link to performance and to different levels of strategy. In general, expected operating performance is an objective of the engineering alternatives. Thus, under the proposed scheme there are two classes to address this weakness: system facets and performance. The four *system facets* are strategy, competence, capacity, and structure. The first three are not emphasized in other enterprise architectures. Notice that structure has been

separated from enterprise elements because structure is contingent on how the enterprise elements are interrelated and grouped. The four *performance* measures are cost, quality, time, and benefit.

Table 7: Classes and Membership Criteria

Classes	Members	Membership Criteria
Enterprise elements	Work Resource Decision Information	Enterprise elements are the parts of interest, the system components.
System facets	Strategy Competency Capacity Structure	System facets relates to setting the nature and intrinsic characteristics of the system. It is the element-facet combination that is called "view" by enterprise architectures.
Engineering activities	Specification Analysis Design Implementation	Engineering activities are phases, equivalent to product development phases, through which the enterprise system is engineered.
Performance	Cost Quality Time Benefit.	These are basic or primitive performance measures that the system is capable of achieving during operations.

All the activities needed for engineering an enterprise system relate to the system's parts of interest, and to how this system is materialized. Hence, only two more classes are needed, namely, enterprise elements and engineering activities. The four *enterprise elements* are work, resource, information, and decision. The confounding of enterprise elements and system facets is called "views" in enterprise architectures. The term view is not used in this research to avoid constraining the framework to information modeling. The four *engineering activities* are specification, analysis, design, and implementation. They are analogous to a life cycle and were synthesized using product development theory.

Three classes in the classification scheme (enterprise elements, system facets, and engineering activities) are concerned with the logical design of the enterprise system, providing the necessary alignment among the enterprise elements and among the system and its environment. These three classes are interdependent and complementary. The fourth class in the classification scheme relates to the desired system performance, and it is dependent on the other three classes. Performance is a function of how well each of the enterprise elements and facets has been engineered.

The proposed classification scheme is shown in Table 8 and Figure 12. Table 8 shows that the classes can be represented by a positional column vector arrangement to convey the message that designing an enterprise system is a process. It is a process that creates value by putting together enterprise elements, uses system theory and considers an enterprise system as the product to engineer, uses engineering activities to make it, and targets expected performance of such product to guide design decisions.

Table 8: The ESE Classification Scheme

Enterprise Element	**Systems** Facet	**Engineering** Activity	**Performance** Measure
Work	Strategy	Specification	Cost
Resources	Competency	Analysis	Quality
Decision	Capacity	Design	Time
Information	Structure	Implementation	Benefit
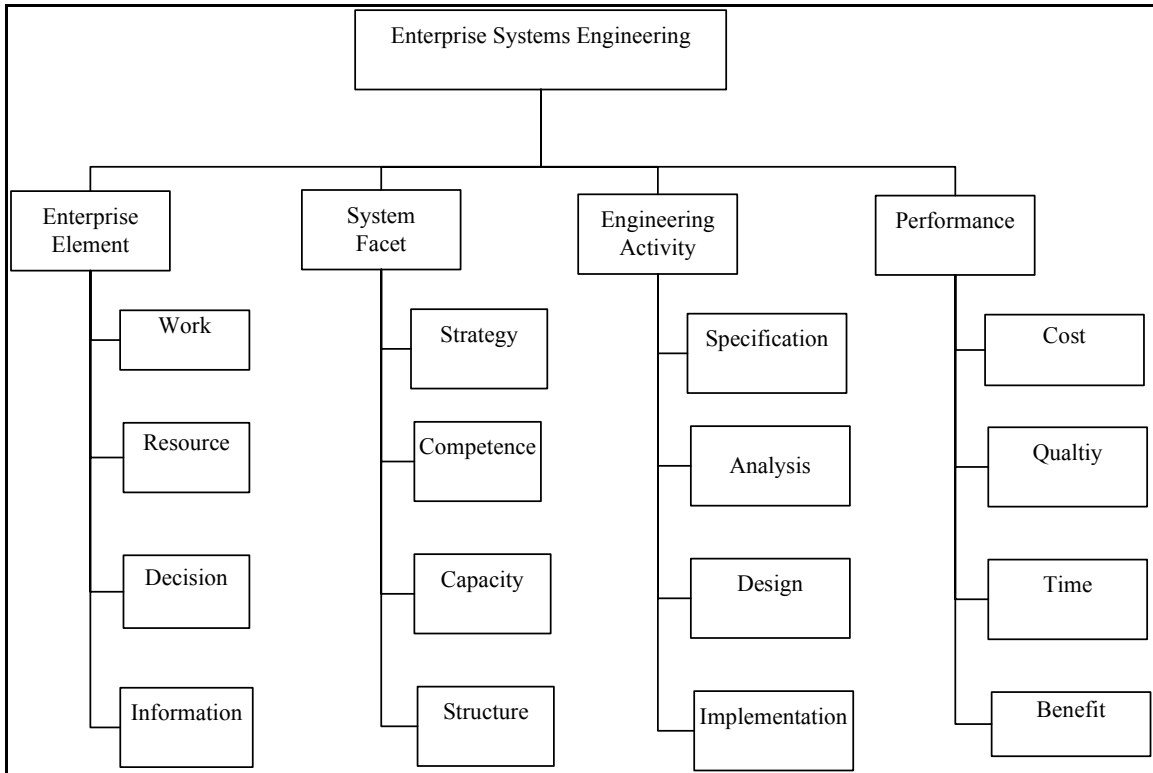
Figure 12: The Classification Scheme

Effective management of an enterprise starts by engineering its business processes and then controlling them (Zelm, 2003). Using the classes in the classification scheme, an enterprise system can be generally described beyond a set of concurrent business processes. An enterprise system is an aggregation of work elements under certain order, rules, and direction given by the decision element. Resources perform work and decisions, and other resources support, are consumed, or transformed by the work. Performing work uses and produces information. An enterprise system is engineered through a life cycle, complies with a strategy, possesses competencies, exhibits flows, has a structure and capacity, achieves certain performance, and has the purpose of producing and delivering a product to a customer. See the general relationships among the four classes in the classification scheme in Figure 13.
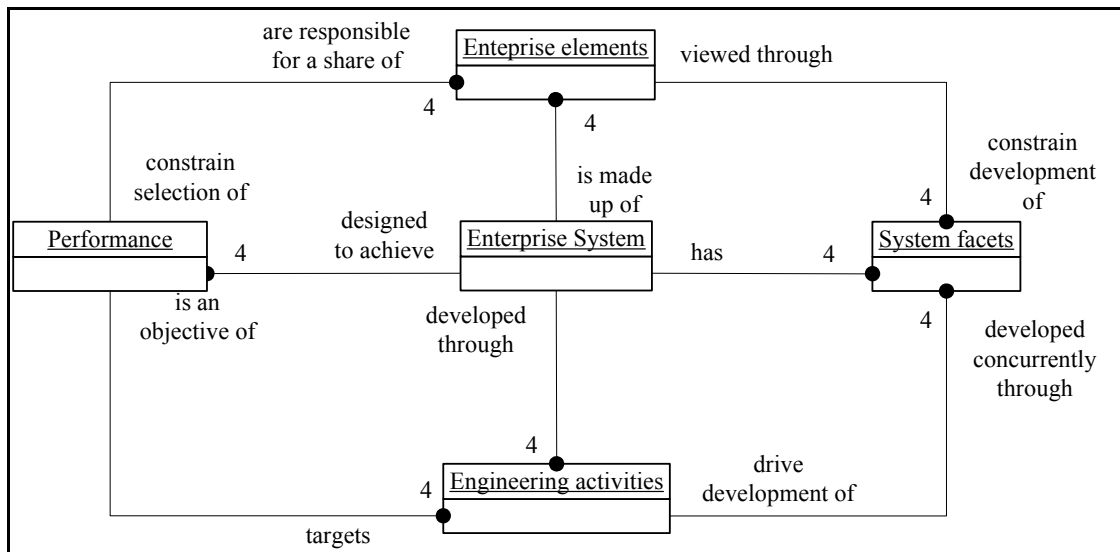
Figure 13: Relationships among Classes in the Classification Scheme

5.2.2   Enterprise Elements

Enterprise elements are the parts of interest, that is, the enterprise system components. Thus, for an object to belong to this class, it has to be a system component of interest. In an enterprise system, there are four main elements: work, resource, decision, and information. The selection of these four enterprise elements obeys two criteria. First, they are present explicitly or implicitly in all the enterprise architectures analyzed in this research, including the information systems architectures ARIS and Zachman's. Second, every object of interest in an enterprise system fits in one of these four subclasses of enterprise elements.

Work (W) is defined as the effort to create and deliver value to customers according to the objectives of an enterprise system. Work consumes energy from resources (e.g. mechanical, kinetic, chemical, biochemical), and it involves the transformation of inputs into outputs according to some specifications. There is a hierarchy of work. The grouping of work

together with the decisions involved to perform it has received several names based on its level of aggregation. For example, a set of work elements is a *task*, a set of tasks is an *activity*, and a set of activities is a *business processes*.

A *resource* (R) is defined as any entity able to perform or support work or decision elements. Resources may also be consumed (e.g. cleaning materials) or transformed by the execution of work (e.g. raw materials). The physical subsystem, one of the main enterprise subsystems according to GIM, results from the implementation of the resource enterprise element. An enterprise system has many types of resources: business units, products, services, markets, customers, intellectual property, facilities, functions, business processes, technology, competencies, and people. PERA classifies resources in three groups: humans, manufacturing, and IT (Williams et al., 1998; Williams, 1999). GERAM considers two different classification of resources: (1) humans and machines and (2) hardware and software (Bernus & Nemes, 1996). Vernadat (1996) classified resources into three groups: humans (e.g. managers, engineers, operators), devices (e.g. IT, manufacturing, logistic), and applications (e.g. off-the-shelf software, in-house built software). However, the proposed ESE classification scheme classifies resources into two groups, active and passive resources (Vernadat, 1996), because of its generic nature and the need to differentiate the resources that perform work and decisions. Active and passive resources are the following:

- Active resources can perform work, decisions or both. There are three main types of active resources: human resources, software applications, and manufacturing machines. A resource in a business process may not necessarily be an employee or an asset owned by a business, because a customer may perform part of the work or decisions in a self service-oriented transaction.

99

- Passive resources are objects being transformed (i.e. raw materials), being consumed (e.g. utilities, other consumable materials), or resources supporting the execution of work (e.g. equipment, tools, facilities, and information and communication technology hardware and infrastructure). A passive resource can be an intangible asset (e.g. intellectual property) used by an organization to develop, manufacture, and deliver products or services to its customers (Coulter, 2002). Goranson (2003) called this type of resources a *second order* resource.

A *decision* (D) is defined as choosing among a set of alternatives. It can be argued that a decision is a subclass of work because it is performed by resources and consumes some energy. However, decisions are placed in a separate class due to a fundamental distinction with the work element: they do not add direct value. Furthermore, decision making requires information inputs only, it does not produce a physical output, and it may affect the nature and existence of other enterprise elements. Another reason for having decision as a separate subclass is to facilitate changes in the work or in the decision element without affecting the other. The latter is similar to what information systems have done in order to separate functionality of an application from the possible information flows. The management subsystem, one of the main enterprise subsystems in GIM and PERA, results from the implementation of the decision as an enterprise element.

The role of the decision element in the ESE scheme is to support the coordination and interactions among all the enterprise elements. The decision state space of an enterprise is defined as the set of potential decisions within the scope delimited by the enterprise's mission and vision. Decisions have hierarchy (e.g. strategic, tactical, and operational).

Decisions may be further classified as either static or dynamic. Decisions handled by automated resources tend to be static, limited to managing changes in volume. Decisions handled by humans tend to be dynamic, involving qualitative changes in objectives or in the nature of the work to be done (Olegario & Bernus, 2003). Others have classified decisions in strategic, management control, and operational control; and in structured and unstructured decisions (Checkland & Holwell, 1998). The combination of work and decision elements result in what is called function by CIMOSA (Kosanke & Zelm, 1999).

Information (I) is being defined as data and knowledge organized to support some work and achieve some business purpose. Data are facts about the enterprise and its everyday transactions from which information is produced (Whitten, Bentley & Dittman, 2001). The final purpose for information is to enable a resource to take the right action. Hence, information is analyzed and interpreted to generate knowledge. In the 1980's, information was considered one of the most valuable assets of an enterprise, but in the 1990's it was realized that knowledge was of more value. Thus, efforts were made to capture knowledge in a knowledge base. This action has led to viewing knowledge as information. Information has a unique attribute: it is not scarce, and it is still available after it has been used (Drucker, 1999). An information system, which embeds data and knowledge, results from the implementation of the information element. Information can be further classified according to its use (e.g. transactional, managerial).

In summary, an enterprise system is made up of four enterprise elements: work, resource, decision, and information. Active resources perform work and use information. Passive resources support, are transformed, or are consumed by the work element. The information

element can be used to represent the other three elements. These relationships among enterprise elements are represented in Figure 14.
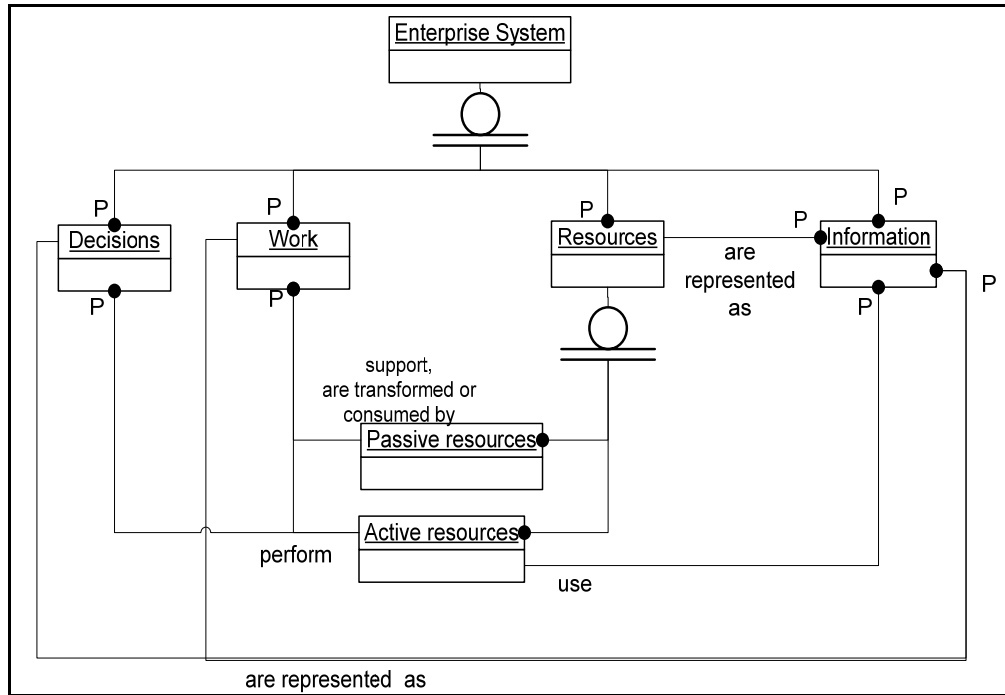
Figure 14: Relationships among Enterprise Elements

5.2.3   System Facets

One distinctive feature of this research is the explicit treatment of the enterprise as a system. Hence, the classification scheme includes a class that enables such treatment. System facets relates to setting the nature and intrinsic characteristics of the system. Thus, an object has to relate to sets of intrinsic characteristic of the enterprise system to be part of this class. An enterprise system has four system facets: Strategy (S_S), Competency (S_C), Capacity (S_K), and Structure (S_O).

Strategy is the creation of a unique, sustainable position, involving sets of enterprise elements and creating fit among them to deliver a unique mix of value to customers and stakeholder (Porter, 1996; Kaplan & Norton, 2000). Strategy must be included in any enterprise design to tie operations to business goals (Goranson, 2003). It is within the strategy realm to decide which properties an enterprise system must exhibit, such as i.e. agility and flexibility (Giachetti et al., 2003). Consequently, strategy becomes a main constraint for engineering an enterprise system.

Strategy sets the enterprise system direction and concept. Strategy serves as a roadmap to build the competencies needed to establish a position in existent or future markets. The enterprise concept addresses the mission, vision, and corporate culture. Mission expresses the enterprise purpose in terms of customers, products, services, markets, technology, growth, profitability, philosophy, public image, concern for employees, strategic alliances, and business processes and competencies to be developed and executed. Vision establishes the position and competencies the enterprise aspires to have in the future (Kalpic et al., 2003).

Long term success depends on core *competencies*, that is, on what a enterprise can do exceptionally well and use it to deliver value to customers (Martin, 1995; Drucker, 1999). Based on Kalpic et al. (2003), Molina (2003), and Collis (1994), the following can be said about competencies: 1) core products are produced with core competencies; 2) a core competency is an aggregation of skills, technologies, knowledge, and other intangible resources that the enterprise uses to design and deploy enterprise elements in a way that produces value for customers and differentiates the enterprise from competitors; 3) a

competency can be seen as the aggregation and coordination of cross functional capabilities; 4) capability is the ability to choose, implement, and exploit enterprise elements with excellence in specific functional areas. A core capability can exist at any point on a value stream and can be used in the creation or production of multiple products or services and deliver value to internal an external customers (Martin, 1995; Kaplan & Norton, 2004).

A flow is a tangible expression of competencies. The existence of flows depends on the resources competencies. For example, if more is manufactured in-house there will be fewer flows to or from subcontractors (we consider subcontracting a virtual resource). Flows can be physical or nonphysical. Flows refer to movement or exchange of enterprise elements. Flows take the form of work flow, resources flow, decision flow, and information flow. Flows can occur within the enterprise or between the enterprise and its environment. Work and decision flows are always attached to resources flows; they cannot exist on their own because a flow implies "movement", and work and decisions cannot transit on their own. Adding that information supports resources the consequence is that flows occur only between resources.

Capacity is the quantity or amount of an enterprise element over a period of time. Capacity may be owned or virtual (subcontracted). Over a specific time period, capacity refers to the amount and type of work to be done, decisions to be made, resources needed to perform productive and managerial work; and the amount and types of information required. This system facet is not emphasized in any other enterprise reference architecture, even though it is a basic input for engineering any kind of enterprise system.

Structure is the result of a conscious design choice of work, resources, information, decision, and their relationships. There can be coordination or subordination relationships resulting from allocating roles, positions, responsibilities, and authorities to active resources.

The relationships between the system facets are shown in Figure 15. An enterprise system is governed by a strategy, which becomes a constraint for engineering the system. An enterprise system exhibits flows and creates value based on its competencies. An enterprise system has a capacity, which in turn depends on the amounts of enterprise elements it is able to manage. An enterprise system has a structure, which represents how the enterprise elements are interrelated and organized.
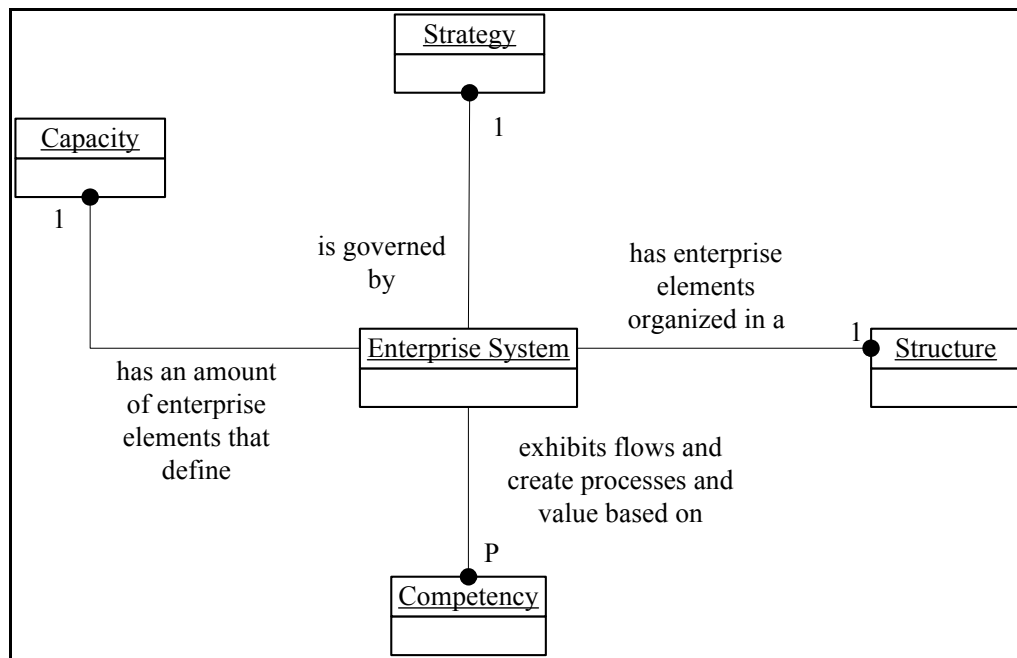
Figure 15: Relationships among System Facets

5.2.4 Engineering Activities

Like other products, an enterprise system is engineered through a life cycle. Engineering activities are phases, equivalent to product development phases, through which the enterprise system is engineered. Thus, for an object to belong to this class it has to be a phase of the life cycle. Enterprise reference architectures (GIM, PERA, and CIMOSA) attempt to show relationships among enterprise elements across the life cycle that puts them together as a system. The proposed engineering activities are analogous to the product development activities of Ulrich and Eppinger (2000): planning, concept development, system-level design, detail design, testing and refinement, and production ramp-up.

The specific engineering activities for product development depend on the final product. When the final product is a physical system, it makes sense to consider implementation as all the necessary processes to actually build it, rebuild it, or change it, such as: acquiring, configuring, testing, validating functionality of components and system, and releasing to operation (IFIP-IFAC, 2003). This type of building is done by other engineering fields (e.g. civil, mechanical, electrical, computer, and software engineering). Including physical construction contributes to the broad scope of enterprise engineering, but does not contribute to a unique identity of this field. The proposed definition of ESE has the specific scope of "developing the models of the coordinated network of business processes – or part of it – that delivers or supports the delivery of value to customers". Consequently, the final product of ESE is a set of design blueprints.

The scope of the engineering activities is established when the enterprise system is in steady-state. In general, a system is in steady-state if it spends a known fraction of time in each of a

set of finite states. If the fraction of time that the system remains in its possible states is changing, the system is said to be in *transient-state* (Ravindran et al., 1987; Hillier & Liberman, 2002). An enterprise system's *steady-state* is defined as a period of time during which there are no changes in its design (i.e. in the design of its network of enterprise elements). Using a similar analogy, an enterprise system is in transient-state when a change in its design is in progress for improvement, divestiture or any other reason. Such a change may be in the enterprise elements (e.g. nature of work performed) or in the systemic facets (e.g. structure, strategy).

Within the scope of the proposed definition for ESE, ESE focuses on the transient state of an enterprise system, that is, ESE supports changes in the enterprise system design. This implies that the operations phase must not be considered in the life cycle of ESE because operational changes, such as the amount of resources or schedules, do not affect the intrinsic design of the enterprise system. Nevertheless, the operations phase is a potential source of initiatives geared toward changing the enterprise system design (e.g. improvement, reengineering). During operations, the enterprise is constantly looking for best practices to extend its uniqueness and productivity (Porter, 1996).

Bounded by the proposed definition of ESE, building upon the complete life cycle of PERA, and guided by the product development phases (Ulrich & Eppinger, 2000), the engineering activities in the ESE classification scheme have been established as in Table 9. The engineering activities are not a sequence; rather, they are iterative to respond to gain knowledge. Further, for existing systems, it is possible to start at an activity different from specification.

The engineering activities in the ESE process are:

- Specification.

- Analysis.

- Design.

- Implementation.

Table 9: Life Cycle Comparison

Proposed ESE Activities	Equivalent to	
	Product Development Phases (Ulrich & Eppinger, 2000)	PERA Life Cycle (Williams et al., 1996)
Specifications	Planning	Identification; concept
Analysis	Concept development	Definition
Design	System-level design, subsystems assembly design Detail design, elements assembly design	Functional design Detail design
Implementation	Testing and refinement: design changes, fabrication an assembly process, training plan, and supplier selection.	Construction Operation and maintenance: not included in ESE Renovation, and disposal and legal dissolution may be considered new ESE projects.

Specification comprises product planning and concept development activities. Product planning includes identification of opportunities; evaluation, prioritization, and allocation of resources and timing to projects; and the mission statement, assumptions, and constraints for the ESE initiative. Concept development includes identifying customer requirements and translating them into system specifications (Ulrich & Eppinger, 2000). Specifications serve as criteria for achieving coordination among the four enterprise elements, objectives, products, and performance. They are a road map providing guidance for transitioning from

the current state to the target or new enterprise state; and for estimating the required investments and recurrent costs for making the transition (OMB, 2004).

From a product design perspective, a *requirement* is any attribute desired in a material, product, process, or system. Requirements are translated into *specifications* which, in turn, are the basis for the development of concept solutions (i.e. product or service). A specification describes what the system has to do, it is a measurable attribute of the final solution, and expresses precisely and unambiguously what will be achieved to address customer and stakeholder requirements (Ulrich & Eppinger, 2000; Suh, 2001). Specifications guide the engineering of the enterprise system and its elements, and are used to evaluate design solutions, hence the label for this activity.

Requirements, and consequently specifications, can be classified as functional and non-functional. Functional requirements directly address the delivery of the main product or service. Nonfunctional requirements vary with the context and refer to other desired properties of the system (e.g. reliability, availability, security, flexibility, maintainability, modularity, ability to integrate, ergonomics, and use of standards). Requirements change over time due to external forces (e.g. market, competition, technology), or internal changes, a reason in favor of having a consistent ESE framework. Requirements may come from customers and stakeholders (e.g. reporting to government agencies, complying with laws, regulations, industry agreements, or environmental constraints). Customer requirements become inputs and stakeholder requirements become constraints of the ESE process.

Analysis focuses on system level. Analysis is based on the input of enterprise system specifications, generates solution concepts, and selects one solution concept for further development in the following engineering activities. A solution concept or conceptual design is an approximation to the future technology, working principles, general configuration of the future system and an approximation of how it will satisfy the customer requirements (Ulrich & Eppinger, 2000). Chen, Vallespir and Dougmeingts (2003) called this activity preliminary design, which constrains the universe of possible final solutions.

Reaching a solution concept in turn requires the investigation of the current situation and relevant internal and external aspects that may influence the enterprise system specifications and design. Analysis includes the identification of competitive environment, industry and economic trends, impact of general policies for the enterprise system design, and investigation of the feasibility of concepts for the overall enterprise system architecture.

Design is defined as the mapping between requirements and a solution that satisfies those requirements. The desired customer attributes are translated into functional specifications. Functional specifications need to be mapped into the enterprise system architecture, subsystems architecture, elements design, their relationships, and flows. It also includes a preliminary integration plan, which is equivalent to a preliminary assembly plan for the enterprise elements (Ulrich & Eppinger, 2000).

In this research, *implementation* is producing the set of design models that represent the network of enterprise elements that create value to customers and their integration, all in accordance with a strategy and customer and stakeholder specifications. Accordingly, the

implementation activity includes a system-wide implementation design, a detailed implementation design for each subsystem (physical, information, and management), a deployment and installation process design, and a training design.

The mapping, checking, and refinement between specifications and the design solution are ongoing tasks during the engineering activities. Implementation requires knowledge of available technological solutions and potential suppliers. Make vs. buy decisions are made and possible alternative solutions are evaluated against the specifications. The best technical solution is selected; specific enterprise elements and the subsystems they are part of are mapped into the refined requirements as part of a validation exercise (Chen et al., 2003). The technical solution is mapped into process variables; process variables are the specifications for the process that will produce the actual enterprise system and its installation (Suh, 2001).

5.2.5   Enterprise System Performance

Druker (1999) stated that at the center of modern society, economy and community is neither technology, nor information nor productivity. It is the enterprise system, or as he called it, "the organ of society" that produces results. An enterprise system produces results during operations. Operational performance has strategic importance because an enterprise must compare itself with industry leaders worldwide.

Operational performance is dependent on the system design (Giachetti et al., 2003). System performance results from design decisions regarding the selected enterprise elements, technology, structure, and competencies. ESE targets some desired system performance, and

uses it as an objective for the enterprise system design and integration (Kaplan & Norton, 1996; Drucker, 1999; Manganelli & Hagen, 2003; Molina, 2003).

System performance is not the sum of the part's performance. Rather, it is the result of their interactions (Patton, 2002). Performance measurement in ESE is focused on the value produced by the system, not its parts. Focusing on the system facilitates improvements in the whole as compared with improvements in the system elements and it enables systemic alignment. Performance measures that facilitate systemic alignment use a single or few primary metrics. Such measures are traceable to the enterprise elements (Manganelli & Hagen, 2003). Four primary performance measures have been identified:

- Cost.

- Quality.

- Time.

- Benefit.

Quality, cost, and time are related to tangible objectives, whereas benefit focuses on more difficult to measure but desirable objectives, such as flexibility (Lim et al., 1997). These basic measures can be hierarchically decomposed and applied to any subsystem or resource having a share of the responsibility for achieving overall system performance. At corporate level, enterprise effectiveness is goal or objectives-oriented (core performance indicators); it takes a financial perspective. At division level, performance (key performance indicators) is focused on the customer. At operational level, performance is process and resource centered (department and personal), and oriented toward efficiency (IFIP-IFAC, 2003).

Cost is directly expressed in monetary terms. Many performance measures are associated with or derived from cost (cost/resource, cost/time). Cost is the basis of many management paradigms, as activity-based costing, which focuses on aggregations of the enterprise element work. Likewise, typical accounting systems reflect cost of resources over time (e.g. operating costs), or amount of resources available (e.g. balance sheet). Cost may be further classified, e.g. variable, fixed, recurrent, capital investments.

The definition of *quality* has evolved over time from conformance to specifications, which is necessary but not sufficient, to meeting customer expectations (Kaplan & Norton, 2004). From an engineering perspective quality is fitness for use, which in turn depends on the interaction of the design and process conformance. The deliberate choices made during design are responsible for the quality of the final product. Process conformance refers to reducing variability and errors in the production process so the final product is consistently manufactured defect-free (Montgomery, Runger & Hubele, 2001).

Time refers to lead time. It is an interval, or period, during which the system provides a response to a customer, such as order fulfillment time, delivery time, or time to market. Time is usually associated with other cells in the classification scheme to derive other performance measures as consumption of resources over time, resource utilization, and cycle time (time/work).

Benefit adds value for customers or stakeholders. It contributes towards the enterprise system goals, objectives, competitive position, or leads to increased revenues or profit, e.g. avoided liability, reduced risks, customer satisfaction, customer retention, build-up

knowledge or competencies, better teamwork, impact control or decision making, impact employee motivation, safety, attract and retain workers, innovation, revenue growth, new revenue sources, environmental safety, personnel health, innovation.

The four enterprise elements, the four system facets, the four engineering activities, and performance can be further divided into additional classes, as mission oriented and support oriented. This third level in the classification scheme is not shown for the purposes of this research.

5.3   Proposed ESE Notation

A feature of the classification scheme is that enables the visualization of areas of study within ESE. ESE's areas of study should be focused on performance, depending on what element to study, which facet of it, and where in the life cycle. The proposed scheme identifies 256 possible areas of study, which raises the challenge of how to compactly identify each one of these areas. In queueing theory, it is customary to use Kendall's notation to identify queueing systems, and its associated queueing models, based on the characteristics of such systems (Ravindran et al., 1987; Hillier & Liberman, 2002). It is being argued that a similar idea can be used to develop an ESE notation that would provide for a compact means of labeling each one of those 256 areas of study, and to later identify models associated with these areas.

Each class in the ESE classification scheme has been given a vector symbol. Each cell in each of the four classes of the classification scheme has been given a unique identifier as shown in Table 10.

Table 10: Notation for ESE Research Areas

Enterprise Element (α)	**Systems** Facet (β)	**Engineering** Activity (γ)	**Performance** Measures (δ)
Work (W)	Strategy (S_S)	Specification (E_S)	Cost (P_C)
Resources (R)	Competency (S_C)	Analysis (E_A)	Quality (P_Q)
Decision (D)	Capacity (S_K)	Design (E_D)	Time (P_T)
Information (I)	Structure (S_O)	Implementation (E_I)	Benefit (P_B)

Let

$\vec{\alpha}$ = vector of enterprise elements $\vec{\alpha} = \{W, R, D, I\}$

$\vec{\beta}$ = vector of system facets $\vec{\beta} = \{S_S, S_C, S_K, S_O\}$

$\vec{\gamma}$ = vector of engineering activities $\vec{\gamma} = \{E_S, E_A, E_D, E_I\}$

$\vec{\delta}$ = vector of performance measures $\vec{\delta} = \{P_C, P_Q, P_T, P_B\}$

Let $\vec{A}$ = collection of ESE areas of study. Then, a 4-tuple can be defined to uniquely identify each area of study:

$$\omega_i = \{(\alpha_j, \beta_k, \gamma_l, \delta_m) : \alpha_j \in \vec{\alpha}, \beta_k \in \vec{\beta}, \gamma_l \in \vec{\gamma}, \delta_m \in \vec{\delta}\}$$

$$\forall \ j, k, l, m = 1, 2, 3, 4$$

$$\vec{A} = \{\omega_i : i = 1, 2, ... 256\}$$

Other collections can also be developed using this notation. For example, a collection of areas studying the impact of the three interdependent classes on performance may be formulated as follows:

Let $\vec{A}_p =$ collection of areas of ESE studying impact on performance; i.e. the performance class is blocked out.

$$\Omega_i = \{(\alpha_j, \beta_k, \gamma_l) : \alpha_j \in \bar{\alpha}, \beta_k \in \bar{\beta}, \gamma_l \in \bar{\gamma}, \delta_m \in \bar{\delta}\}$$

$$\vec{A}_P = \{\Omega_i : i = 1, 2, ...64\}$$

Another example would be to focus on studying the effect on performance of each of these independent classes.

Let $\quad \vec{A}_E =$ ESE areas studying impact of enterprise elements on performance.

$\vec{A}_F =$ ESE areas studying impact of system facets on performance.

$\vec{A}_A =$ ESE areas studying impact of engineering activities on performance.

$$\vec{A}_E = \{(\alpha_j) : \alpha_j \in \bar{\alpha}\} \quad j = 1,...4$$

$$\vec{A}_F = \{(\beta_k) : \beta_k \in \bar{\beta}\} \quad k = 1,...4$$

$$\vec{A}_A = \{(\gamma_l) : \gamma_l \in \bar{\gamma}\} \quad l = 1,...4$$

Note that $\vec{A}_E \subset \vec{A}$, $\vec{A}_F \subset \vec{A}$, $\vec{A}_A \subset \vec{A}$, and $\vec{A}_P \subset \vec{A}$. These collections are useful for two purposes: (1) to describe areas to addresses within ESE and to classify other research efforts related to the subject of ESE and (2) to guide the enterprise engineering process. The targeted performance becomes a constraint in the ESE process, as will be shown in the next chapter, because system performance is a function of design decisions, $\delta = f(\alpha, \beta, \gamma)$. Hence, collection $\vec{A}_p$ is bound to have significant value in both, industrial applications and research.

The concatenation operator (//) can be applied to the members of collections $\vec{A}$ and $\vec{A}_P$ to produce a descriptive statement of the areas. For example:

(W // S_S // E_D // P_T) $\rightarrow$ work strategy design and its impact on time, α=W, β=S_S, γ=E_D,

and δ=P_T.

(I // S_O // E_A // P_C) $\rightarrow$ information structure analysis and its impact on cost, α=I, β=S_O,

γ=E_A, and δ=P_C.

(W // S_S // E_D) $\rightarrow$ work strategy design and its impact on performance, α=W, β=S_S,

γ=E_D.

Similarly, predicate logic could be used to derive these descriptive statements. Given the predicate:

Impact-on ((α, β, γ)), δ) $\wedge$ Impact-on ((α, β, γ), performance)

The following instances of the predicates are equivalent to the previous examples:

Impact-on ((W, S_S, E_D), P_T)

Impact-on ((I, S_O, E_A), P_C)

Impact-on ((W, S_S, E_D), performance)

The ability to generate these descriptive statements enables a consistent terminology and enhances understanding.

5.4 Validation of the ESE Classification Scheme

In general, the classification scheme is logically correct. Reasoning, previous research, and empirical experience provided the grounds for establishing the classes and subclasses. A sound pattern of analysis was followed. Theory backed the grounds of any claim and

proposal (Toulmin, Rieke & Janik, 1979). Moreover, this research followed a basic requirement for scientific inquire: describing the procedures used to carry out this study (Gay & Airasian, 2000). Furthermore, another basic validation requirement was met: all the classes in the classification scheme have a counterpart in the empirical world (Xia, 1999).

To further prove its validity the classification scheme was checked in the following three ways:

- Compliance with accepted criteria for enterprise architectures.

- Compliance with scientific criteria for formulating classifications.

- Completeness, which was checked in three ways: by internal homogeneity and external heterogeneity; by showing that the classification scheme subsume other recognized enterprise reference architectures; and by classifying other research efforts.

Regarding *compliance with criteria for enterprise architectures*, Berio and Vernadat (1999) stated that all general architectures, whether accepted or under discussion at the international level, show that any approach for enterprise modeling must at least address three types of flows (material, information, and decision), three modeling levels (requirements, design specification, and implementation description), and four modeling views (function, information, resource, and organization). The proposed classification scheme complies with all these. It addresses material, information, and decision flows by the competency system facet. It addresses the three modeling levels in the engineering activities: specification, design, and implementation. It addresses views in the enterprise elements: information and resource. It addresses the function view by the elements work and decision. Organization is

addressed by structure, contributing a new insight by considering structure a system facet separated from enterprise elements.

The classification scheme is correct when tested against the criteria for developing classifications; specifically:

(1) It has *more than two* classes (Table 10).

(2) There is only *one basis for division* (Beardsley, 1966); each class and subclass is completely distinguishable from one another. Every object of interest in an enterprise system can be classified in one and only one of the subclasses.

(3) The classification scheme is based on *fundamental distinctions among classes* (Hempel, 1965). The four classes (enterprise elements, system facets, engineering activities, and performance) are four fundamentally distinct classes because each one has essential and unique attributes: enterprise elements represent all the objects that make up the enterprise system, system facets provide the requirements and constraints for the system behavior, engineering activities are equivalent to a life cycle for product development, and performance represents the expectations that the system will satisfy once in operation.

(4) The classification scheme has *membership conditions* (Hempel, 1965), as each subclass is specified so that there are clear classificatory concepts for the objects being studied (Table 7).

(5) The *relationships among classes* (Beardsley, 1966) are established in the form of IDEF1x diagrams (Figures 12, 13, and 14).

(6) The classification scheme has a *hierarchy*, which allows further levels of division. Three levels are required for a scientific classification (Gay & Airasian, 2000).

Regarding *completeness*, all the objects that make the enterprise system, within the definition for ESE, belong to one of the enterprise elements. The classes in the classification scheme are internally homogeneous; the objects within one class belong together, unlike other reference architectures that confound elements with system perspectives. The classes are also externally heterogeneous. The four classes in the classification scheme are clearly different and represent the intended whole, which was tested by the absence of unassignable objects that belong to enterprise systems (Patton, 2002). The classification scheme subsumes the architectures proposed by PERA, CIMOSA, and GIM up to the implementation activity. This was shown by the comparative analysis in section 5.2.1.

A last check for completeness was based on a validation approach by Gay and Airasian (2000), in which the categories in a classification derived from a subset of data were applied to a second set of data to check if the categories held up for the second set. The categories in the classification scheme were mainly developed by benchmarking, using other reference architectures, and by the author's empirical experience. Further, the classification scheme has been used to classify previous research efforts. In order to classify other's research, the literature used to support this research was input into an EndNote bibliographical database. It was then analyzed using a SPSS database where four variables were created, one for each class: enterprise element, system facet, and engineering activity. Each variable can have one of four possible values, one value for each subclass, e.g. the variable for enterprise elements can have the four values: work, resource, decision, and information. Depending on its main content, each research paper in the area of ESE was assigned four values. The number 3 in italics and bold in Table 11 indicates that 3 of the research papers in the sample dealt with information structure design.

Table 11: Research Classification

Enterprise Element	System facet	Engineering activity				Total
		Specification	Analysis	Design	Implementation	
Work	Strategy	4				4
	Structure	12		3		15
	Competency/ Flow	2		1		3
Resources	Structure	3	1			4
Decision	Structure			1		1
Information	Strategy	4	1		1	6
	Structure	6		*3*		9
	Competency/ Flow	2		2		4
The four elements	Strategy	1				1
	Structure	11	1	1		13
	Competency/ Flow	1		1		2
TOTAL		25	3	12	1	62

A sample of sixty two papers was used in the analysis. Qualitative studies usually work with small, purposive, and theory-driven samples (Miles & Huberman, 1994).   A purposive sample aims at getting insight and offers useful manifestations of the phenomenon under study.   In the case of purposive sample, sampling is terminated when the saturation or redundancy point is reached and no more new information is drawn from increasing the sample size (Patton, 2002). This research considered a large body of research in the area. After classifying more than sixty papers it was decided to stop this validation exercise because the redundancy point was reached. All research within the scope of ESE can be categorized using the proposed classification scheme.

In summary, validity of the classification scheme has been demonstrated by reasoning, by backing it with existing theory, by describing the procedures used to carry out this research, by having a counterpart in the empirical world for each class and subclass, by compliance with accepted criteria for enterprise architectures, by compliance with scientific criteria for formulating classifications, and by its completeness.

CHAPTER VI

THE ESE PROCESS

The objective of the ESE process is to guide the making of a final product: a set of designs of the enterprise system suitable for implementation. This chapter offers:

- Criteria for creating an ESE process.

- A representation of the ESE process using IDEF0 models and Petri nets together with mathematical expressions based on the classification scheme to ensure its completeness.

- A description of the main activities and sub-activities of the ESE process. Details of the lower level activities are provided in the Appendix, where an IDEF0 model shows their input, output, mechanism, and control (ICOM) relationships.

6.1   Specifications for an ESE Process

From the literature review, a process is a set of activities whose execution is guided by rules and triggered by some event, and produces an observable or quantifiable result for a defined customer. Considering that the final product of an ESE process is a set of design blueprints for an enterprise system, the classification scheme, previous work on enterprise engineering and strategy, and product design theory, an ESE process must comply with the following specifications:

1) It must have a product life cycle orientation.

2) It must enable enterprise integration.

3) It must enable the acquisition and satisfaction of customer requirements.

4) It must enable strategy alignment.

5) All models of the process must adhere to the modeling principles criteria (Vernadat, 1996).

Similar to enterprise engineering methodologies, an ESE process must encompass the enterprise system *life cycle* in the form of process models or structured procedures (IFIP-IFAC, 2003; Bernus & Nemes, 2003). Hence, it must include activities for planning, concept development, system-level design, detail design, fabrication and assembly, and installation (Ulrich & Eppinger, 2000). To enable enterprise *integration* the ESE process must consider the interplay or interactions between enterprise elements: work, resources, decision, and information (Suh, 2001). Desirable interactions are defined as those that lead to an improvement in systemic performance. Also, there must be an ordered linking between enterprise elements and system facets through the engineering activities because there must be integration among different enterprise orientations or perspectives (Aguilar-Savén, 2002a). *Customer requirements*, which are translated into functional requirements, must drive the entire process (Ulrich & Eppinger, 2000; Suh, 2001). *Strategy alignment* is used to create fit among activities and to combine activities so that they reinforce one another (Porter, 1996). The resulting network of enterprise elements must support the enterprise strategy (Ortiz et al., 1999).

Based on previous work, Vernadat (1996) proposed a set of criteria for developing enterprise models that he called *modeling principles*. These criteria address necessary issues for enterprise system modeling; hence, the model that represents the ESE process must also adhere to these criteria:

- There must be a purpose: enterprise models address some desired finality, such as understanding, knowledge reusing, analyzing, designing, redesigning, simulating, decision-making, controlling, coordinating, or monitoring.

- There must be a clear scope and domain.

- There must be a viewpoint.

- It must have a defined level of granularity.

- It must enable functional decomposition.

- It must enforce modularity and reusability.

- It must decouple functionality (what to do or work) and behavior (how to do it), processes and resources, and data and control.

6.2 Proposed ESE Process

The ESE process is intended to guide the engineering of an enterprise system. Engineering an enterprise system means specifying, analyzing, designing, and implementing the blueprints of the network of elements that produce products or services of some value to its customers while meeting required performance. Integration of the whole enterprise and effective use of resources requires knowledge of the business processes, their interactions, resource capabilities, the goals of the enterprise and the goal of each business process (Kosanke & Nell, 1999a). The proposed ESE process allows for applying such knowledge, managing interactions, and using the enterprise strategy and desired performance as constraints. Hence, part of the challenge is to devise a process that is driven by the engineering activities defined in the classification scheme so that one activity provides inputs to the next and provides feedback to the previous.

Figure 16 shows the 64 areas identified by the classification scheme. Alternatively, Table 12 exhaustively shows the sixty-four areas targeting some pre-specified performance – not shown in the graph – in a two dimensional format. Further, Table 12 suggests the need for integration (i.e. physical, data, and application) among enterprise elements. The set of sixty-four areas represent an approach to completely address the engineering of an enterprise system and provides a mechanism to achieve alignment. Each area is a set of activities that must be carried out using appropriate tools.
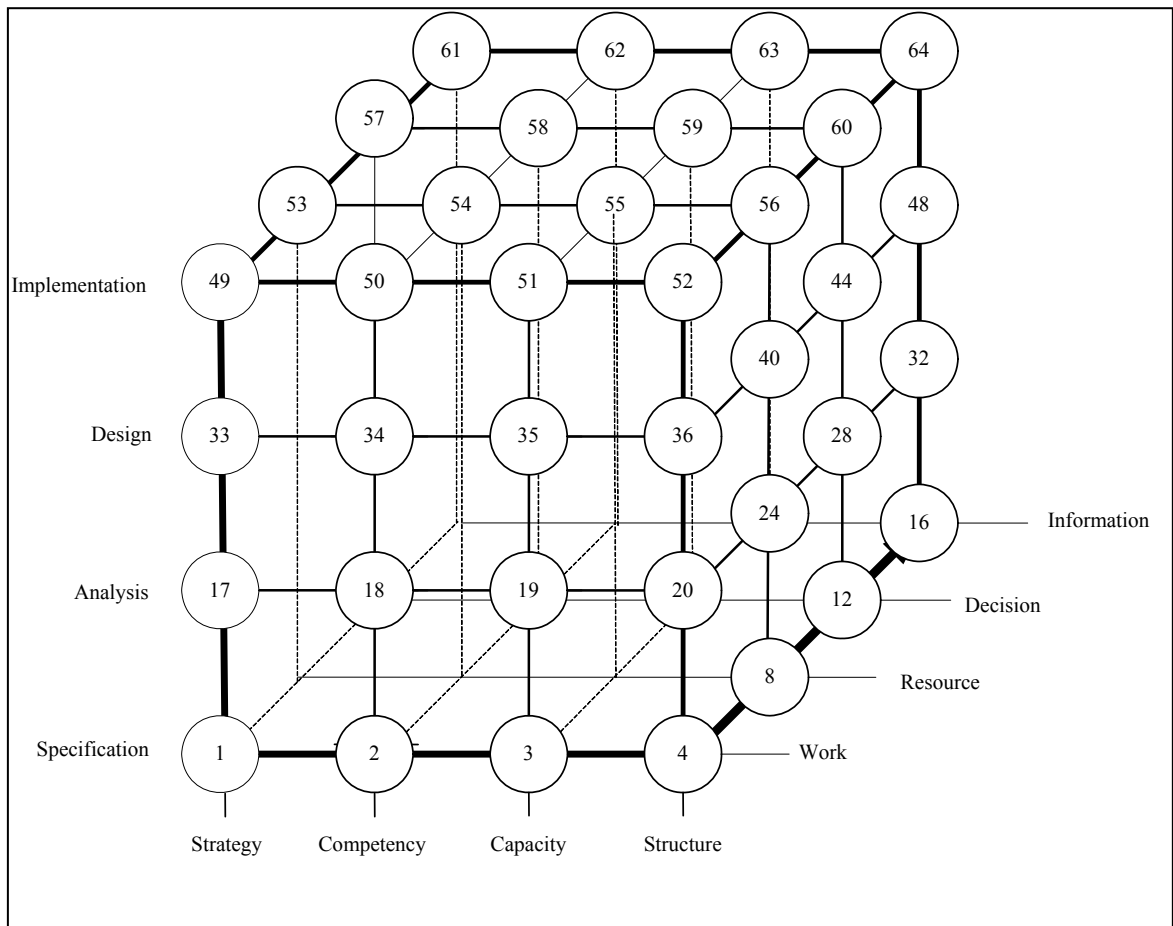
Figure 16: Areas within ESE

Table 12: Illustration of the Areas within ESE

Specification		Strategy	Competency	Capacity	Structure
	Work				
	Resources				
	Decision				
	Information				

Analysis		Strategy	Competency	Capacity	Structure
	Work				
	Resources				
	Decision				
	Information				

Design		Strategy	Competency	Capacity	Structure
	Work				
	Resources				
	Decision				
	Information				

Implementation		Strategy	Competency	Capacity	Structure
	Work				
	Resources				
	Decision				
	Information				

In addition to the activities, a process must have a means to trigger those activities. For an ESE process, such means are events that occur when a particular activity has been completed, or when stimulus, such as feedback, fires the activity. The actual means and stimulus would depend on the specific enterprise system being designed. The ESE process being proposed will have specific inputs and outputs for each of the 64 sets of activities, and it will also have clear channels of feedback and parallelism among all these sets. The best way to present the proposed ESE process is using Petri nets and IDEF0 models. Petri nets represent the process at a macro level, whereas IDEF0 models provide more details inside each set of activities.

The ESE process prescribes *activities* and their interrelationships. The ESE process has been devised so that it allows for integration at several levels, and the resulting enterprise system

is an actual coordinated network of enterprise elements. The ESE can start at any engineering activity; nevertheless, the ICOM interrelationships strive to achieve integration. Interactions are managed by the concurrent treatment of the four enterprise elements across the system facets, providing a way for analyzing connectivity, data sharing, and interoperability; and by the concurrent treatment of the system facets across the engineering activities, providing a way for analyzing coordination, aligning the enterprise design with strategy, stakeholders' requirements, and having all the system components aiming towards common objectives.

6.2.1   Activities of the ESE Process

6.2.1.1 Specification Activity and Sub-activities

The *specification* activity defines what customers consider value (Drucker, 1999). This activity transforms customer and stakeholder needs into specifications, sets the scope of the ESE project, and identifies stakeholders, process owners, and users. The scope of an ESE project includes setting up the boundaries of the system to be engineered, or changed, together with expectations of capacities and capabilities for the system under analysis and the possible constraints for changing such system. Specification encompasses four interrelated sub-activities: strategy specification, competency specification, capacity specification, and structure specification.

Strategy Specification develops corporate and business strategies. Strategy specification is concerned with articulating strategy in the form of the enterprise mission, vision, objectives, value statements, core products, desired core competencies, key success factors and performance indicators, and the identification of necessary resources. Strategy specifications

provide criteria to align the network of enterprise elements among themselves, and the enterprise system with its environment in order to achieve the enterprise medium and long-term objectives. In terms of product development, after identifying, evaluating, and prioritizing opportunities, strategy specification provides a target system, its assumptions and constraints. Strategy specification has four activities: work strategy specification, decision strategy specification, resource strategy specification, and information strategy specification (see Appendix).

Competency Specification identifies competencies needed. The specification of new competencies sets the scope for the design of the enterprise elements in the future system and their corresponding integration needs. A unit of elemental competency, a capability, is the building block to place sets of work and decision elements together (e.g. work flow) and perform business tasks, activities or entire business processes. A flow can be seen as the result of an aggregation of competency units. The aggregation of competency units can be detected by the paths followed by physical (e.g. inventories, work in process) and information objects. Physical and information flows occur among resources. Work and decision flows occur via either material or information elements. Competency specification has four activities: work competency specification, decision competency specification, resource competency specification, and information competency specification (see Appendix).

Capacity specification identifies the enterprise's capacity gaps based on vision, strategy, and competency. It identifies required new capacity and a conceptual solution approach. It defines the size of the system in terms of its throughput or output per time unit based on

forecasts for products and services and the strategy specification. Capacity specification has four activities: work capacity specification, decision capacity specification, resource capacity specification, and information capacity specification (see Appendix).

The *structure specification* guides the actual setting up of the four enterprise elements that deliver the required products and services. An enterprise is a vast system of interdependent components working together to produce value (Manganelli & Hagen, 2003). Strategy specification is the main input for the structure specification (Drucker, 1999) followed by competency and capacity specification. The structure specification includes: (a) the identification of a potential target organizational structure, which may follow the taxonomy of Mintzberg, i.e. machine bureaucracy, professional organization, entrepreneurial, or adhocracy; (b) the expected balance among control, autonomy, and cooperation among resources; and (c) establishing the organizational design principles to follow, i.e. specialization, coordination, knowledge and competence, and control and commitment (Bernus, 2003; Mintzberg, 1979; Keidel, 1995; Goold & Campbell, 2002). Structure specification has four activities: work structure specification, decision structure specification, resource structure specification, and information structure specification (in Appendix).

6.2.1.2 Analysis Activity and Sub-activities

Enterprise system analysis has the enterprise specifications as input. Enterprise system analysis focuses on system level solutions and the possible general configuration of the system, constraining the universe of possible final solutions without considering available components from the market. Enterprise system analysis assesses the gap between the current state and the desired state of the enterprise system in terms of its internal and external

environment, which may influence the enterprise system specifications and design. It also includes the identification of subsystems and the characterization and evaluation of the existing or planned network of enterprise element against the enterprise specifications and target performance (Upplington & Bernus, 2003). Enterprise system analysis has four activities: strategy analysis, competency analysis, capacity analysis, and structure analysis.

Strategy analysis establishes the current and desired states of the enterprise as they relate to the enterprise elements and system facets. Mission defines strategy (Drucker, 1999). Strategy analysis establishes market conditions, stage of evolution (emerging market, established, eroding, erupting market) and trends, product and resource concepts, and the current state of the enterprise. Strategy analysis is the basis for validating the general strategic direction and generic strategy proposed in the strategy specification. It studies the enterprise internal and external environment over which the future functional strategies will be based. Strategy analysis has four activities: work strategy analysis, resources strategy analysis, decision strategy analysis, and information strategy analysis (see Appendix).

Competency analysis establishes the required new competencies at system and subsystem level, establishes the gap with existing competencies, and proposes bundles of competencies. Competency analysis identifies solution approaches such as cultivating, co-developing, licensing or outsourcing competencies. For existing enterprise systems, competency analysis starts by decomposing the flows of enterprise elements to pinpoint elementary capabilities. Competency analysis investigates current and required flows of enterprise elements and checks their feasibility in terms of available resources. Wherever there is a material or information flow, there is an interaction between the enterprise elements handling that flow,

hence, there is a need for coordination or interoperability between the resources involved in that flow. Flows should be mainly the result of the enterprise system design in order to make coordination as efficient and effective as possible. Competency analysis has four activities: work competency analysis, resource competency analysis, decision competency analysis, and information competency analysis (see Appendix).

Capacity analysis consists in evaluating the state of the system to satisfy the specifications in terms of the required amount or quantities of resources, work, information, and decisions. It identifies a solution approach toward capacity, such as aggregated planning or outsourcing capacity. Capacity analysis has four activities: work capacity analysis, resources capacity analysis, decision capacity analysis, and information capacity analysis (see Appendix).

Structure Analysis deals with aggregating enterprise elements. For existing enterprise systems, structure analysis evaluates the current organization structure against specifications, proposes conceptual solutions for the enterprise organization (e.g. job shop vs. a flow shop), and evaluates the feasibility of migrating from one alternative to another. Structure analysis has four activities: work structure analysis, resources structure analysis, decision structure analysis, and information structure analysis (see Appendix).

6.2.1.3 Design Activity and Sub-activities

Designing an enterprise system is designing a network of interacting enterprise elements that produce and deliver value to customers (Molina, 2003). Enterprise system design starts at system level, goes to subsystem design, and finishes at elements design and integration

design. Integration for enterprise system design is equivalent to assembly design for products (see Table 13).

Table 13: Breakdown of the Enterprise Design Activity

Phase	Enterprise Elements	Enterprise Subsystems
System level, logical or concept design	Identification of work, resources, decision, and information classes and interaction.	Coordination within and between subsystems (physical, information, and management). Alignment with system specifications.
Assembly design	Connectivity among resources	Data sharing. Interfaces among resources. Alignment among subsystems
Component level design	Resources & technology selection.	Resources interoperability.

According to the product design theory of Suh (2001), an interaction between different domains (e.g. customer domain, functional domain, physical domain) is needed to manufacture a product. As compared with product engineering, a major complication arises in the engineering of an enterprise system due to a large number of interactions among enterprise elements. Interactions can cause that changes in one enterprise element affect others. The first step to handle interactions is that functional specifications need to be decomposed into a hierarchy of specifications, and these specifications mapped into design parameters (Suh, 2001). The capabilities, capacities, and structure of these three subsystems support the work element. This is still architectural and functional design, decomposed at lower levels until defining how the subsystems and enterprise elements interact with each other. The enterprise design encompasses four interrelated activities: strategy design, competency design, capacity design, and structure design.

Strategy Design formulates operational strategies. Strategy design translates the strategy specification into operational (functional) strategies, such that the corporate and competitive strategies are achieved. Functional strategies must be consistent among them, internally aligned with corporate and business strategy, and externally aligned with the environment. The cash contributions of the portfolio of products are identified. Devising functional strategies and their alignment is the main outcome of strategy design. Strategy design encompasses four interrelated activities: work strategy design, resources strategy design, decision strategy design, and information strategy design (see Appendix).

Competency Design addresses how to develop a balanced portfolio of competencies, a competency acquisition agenda, and is the place where individual enterprise elements and subsystems are actually designed. Competency designs also deals with the resulting flows from those capabilities, and their starting and ending events. Competency design encompasses four interrelated activities: work competency design, resources competency design, decision competency design, and information competency design (see Appendix).

Capacity Design addresses how to get the required capacities, by acquisition, agreements with a supply chain, extended, or virtual enterprises. Capacity design encompasses four interrelated activities: work capacity design, resources capacity design, decision capacity design, and information capacity design (see Appendix).

Structure Design comprises the organization of the enterprise system, its subsystems, and their elements. Based on the enterprise specifications and structure analysis, structure design defines how the enterprise elements will be grouped together, and how to align the

management need for control with the autonomy and cooperation required to develop complex activities (Keidel, 1995). Structure design plays the role of an assembly plan for the system, subsystems, and enterprise elements. Organizational design principles guide this alignment and the general aggregation of the enterprise elements. As proposed by Goold and Campbell (2002) organizational design principles are: specialization, coordination, knowledge and competence, and control and commitment. Applying specialization results in an organizational structure with many units oriented towards specialized work. The coordination principle is decision oriented, favors centralization of decisions and tends to create few units, and tradeoffs must be made with the specialization principle. Knowledge and competence is oriented toward resource competency. Control and commitment is decision oriented, attempting to distribute the power, or the responsibility for the creation of decision frameworks of the organizational units; innovation and adaptation, this principle pushes toward the ability to reconfigure and redeploy the enterprise elements.

Structure design encompasses four interrelated activities: work structure design, resources structure design, decision structure design, and information structure design (see Appendix).

6.2.1.4 Implementation Activity and Sub-activities

Enterprise design involved the generation of feasible alternatives and the selection of one alternative for implementation (Chen et al., 2003). Implementation produces all the necessary blueprints for the actual implementation, called implementation designs, such that the enterprise elements perform their roles in a coordinated fashion according to specifications. Implementation designs specify how to realize the design, system-wide, subsystem-wide and enterprise element-wide. Implementation is divided in process design,

assembly design, and deployment design. Process design is the how to produce the system and subsystems. Assembly design specifies in detail how to put them together, how to integrate them, i.e. network, data, and interoperability or resources. Deployment design specifies procedures for installation, operation, and training (Ulrich & Eppinger, 2000; Whitten et al., 2001; Chen et al., 2003). Enterprise implementation encompasses four interrelated activities: strategy implementation, competency implementation, capacity implementation, and structure implementation.

Strategy Implementation deals with devising specific action plans to serve customers, to compete and operate. Strategy implementation provides a way to actually transit from the current state to the target state of the enterprise. Strategy implementation defines how to evaluate strategy using performance indicators (key, division, project, department and personal performance indicators) to control results and progress toward the defined objectives. Strategy implementation encompasses four interrelated activities: work strategy implementation, resources strategy implementation, decision strategy implementation, and information strategy implementation (see Appendix).

Competency Implementation develops the design of a production process to produce the enterprise system and subsystem; this is an implementation design for securing and realizing required work competencies, decision competencies, resource competencies, and information competencies. It is not required to own all the capabilities needed to manufacture a product; they can be outsourced. Core competencies, used for competing, are developed in-house while other competencies can be outsourced. Elementary competencies are linked with resources with a specific skill and knowledge, and they can be reconfigured to devise

business processes. Competency implementation encompasses four interrelated activities: work competency implementation, resources competency implementation, decision competency implementation, and information competency implementation (see Appendix).

Capacity Implementation develops action plans for meeting time-phased capacity requirements. Capacity implementation integrates the resources that perform or support work and decisions, and it is focused on physical connectivity, data integration, and applications interoperability to achieve a desired performance. Capacity is dependent of resources; business process and the enterprise itself do not physically exist without them. Resources form the physical system, the outer and observable layer of the enterprise system that performs all the work and transforms inputs (e.g. customer requests, information, and materials) into products or services. Capacity implementation has four activities work capacity implementation, resource capacity implementation, decision capacity implementation, and information capacity implementation (Appendix).

Structure Implementation is part of the enterprise engineering process (Bernus, 2003). Organization structure is shaped by a company's strategy, competency and capacity. In general, enterprise structures have certain common components, as described by Mintzberg (1992): strategic apex, techno-structure, support units, middle line, and operations. When several enterprises collaborate, they tend to form a network-like structure (Bernus, 2003). For an enterprise system, structure implementation is analogous to an integration and deployment plan. An assembly design states how to integrate and organize sets of enterprise elements, and assigns roles, authority and responsibility to specific human resources over those sets of enterprise elements. Structure implementation has four activities: work

structure implementation, resource structure implementation, decision structure implementation, and information structure implementation (see Appendix).

6.2.2 Petri Net Models

Petri Nets can be used to represent the 64 set of activities. Petri nets enable a hierarchical macro level process. Three macro level Petri nets were developed. Each activity has been labeled using the proposed ESE notation. Solid bars represent activities. Circles represent the state of the ESE process. The completion of an activity fires the activity that follows. The states are equivalent to the deliverables of each activity (e.g. completed plan, completed analysis).

Figure 17 shows the top level model of the ESE process, clearly driven by engineering activities. The initial ready state means that a decision has been made to fire the ESE process. Unfolding one activity, for instance the specifying activity in Figure 17, renders four sub-activities as shown Figure 18, where "integrating specifications" means to coordinate and perform trade-offs among the strategy specifications, competency specifications, capacity specifications, and structure specifications. Figure 18 is a subset of the level 2 Petri net model. Similar graphs exist for other activities (analysis, design, and implementation). Each one of the activities in level 2 can be further unfolded, yielding a level 3 Petri net model. For instance, Figure 19 shows the graph for strategy specification. In Figure 19, "integrating strategy specifications" refers to managing interactions among work strategy specifications, resources strategy specifications, decision strategy specifications, and structure strategy specifications. Replicating the graph for each activity

138

in level 2 would yield the 64 activities to produce an integrated enterprise system. Feedback

loops from one activity to the previous have been omitted for simplicity.
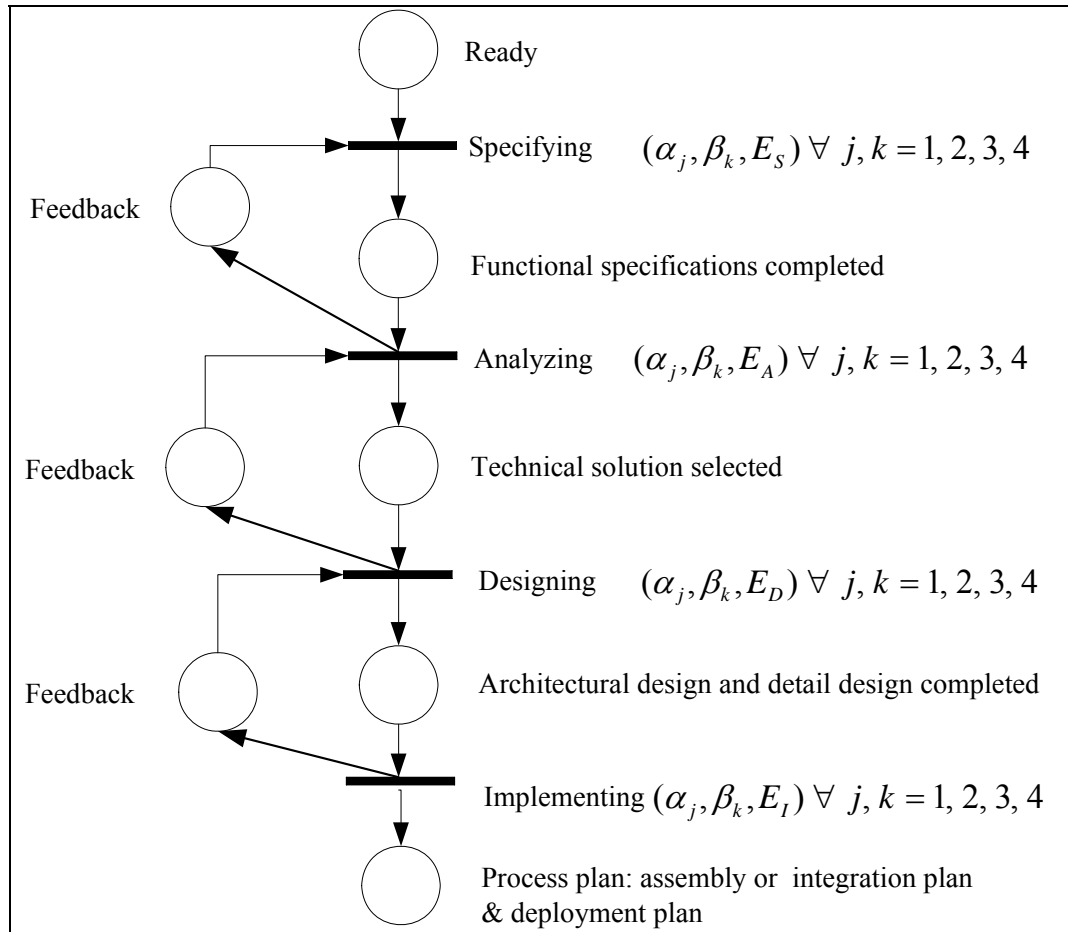
Figure 17: Petri Net Model of the ESE Process
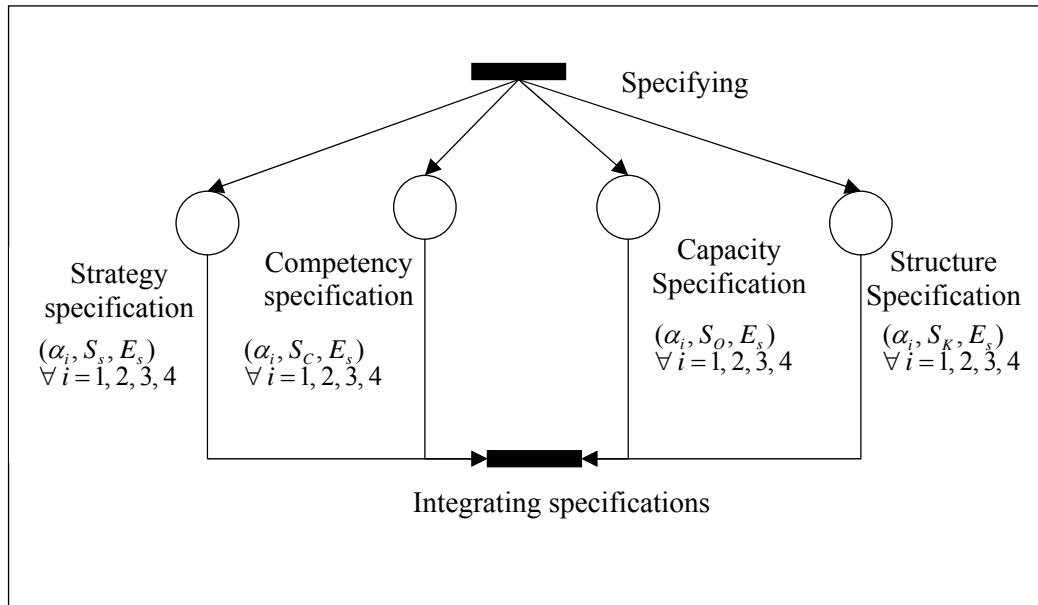
Figure 18: Unfolding the Specification Activity
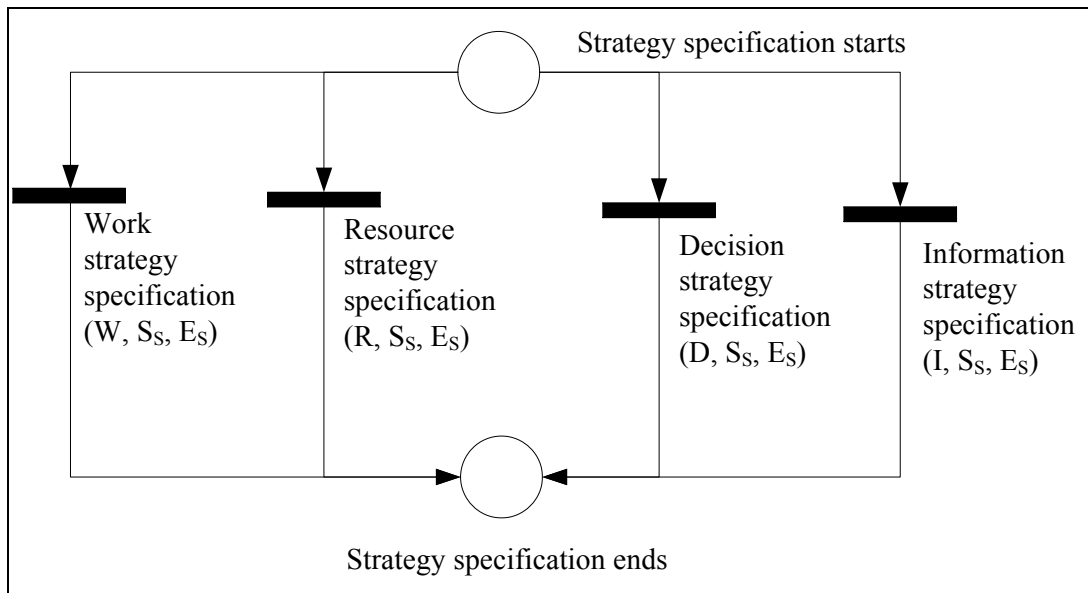
Figure 19: Unfolding the Strategy Specification Activity

6.2.3 IDEF0 Model

To further illustrate the ESE process and show how the outputs of one activity constrain or serve as input for others (ICOM relationships), an IDEF0 model (Figure 20) was developed. Level 1 of this IDEF0 MODEL (Figure 21) addresses the four engineering activities that drive the ESE process. Level 2 addresses the 16 sub-activities, and level 3 addresses the 64 sub-sub-activities. Levels 2 and 3 are presented in the Appendix.
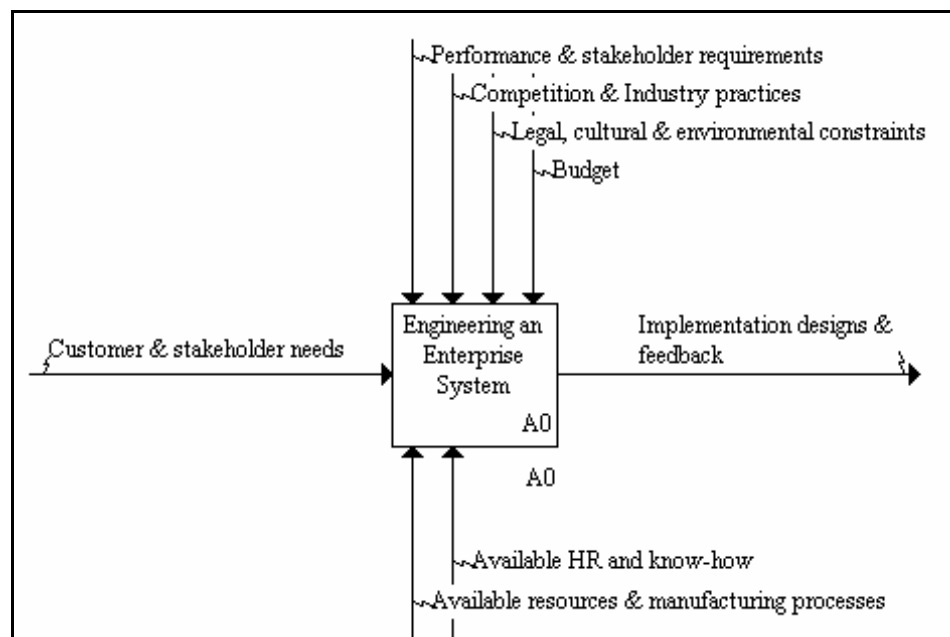
Figure 20: Activity Model of the ESE Process
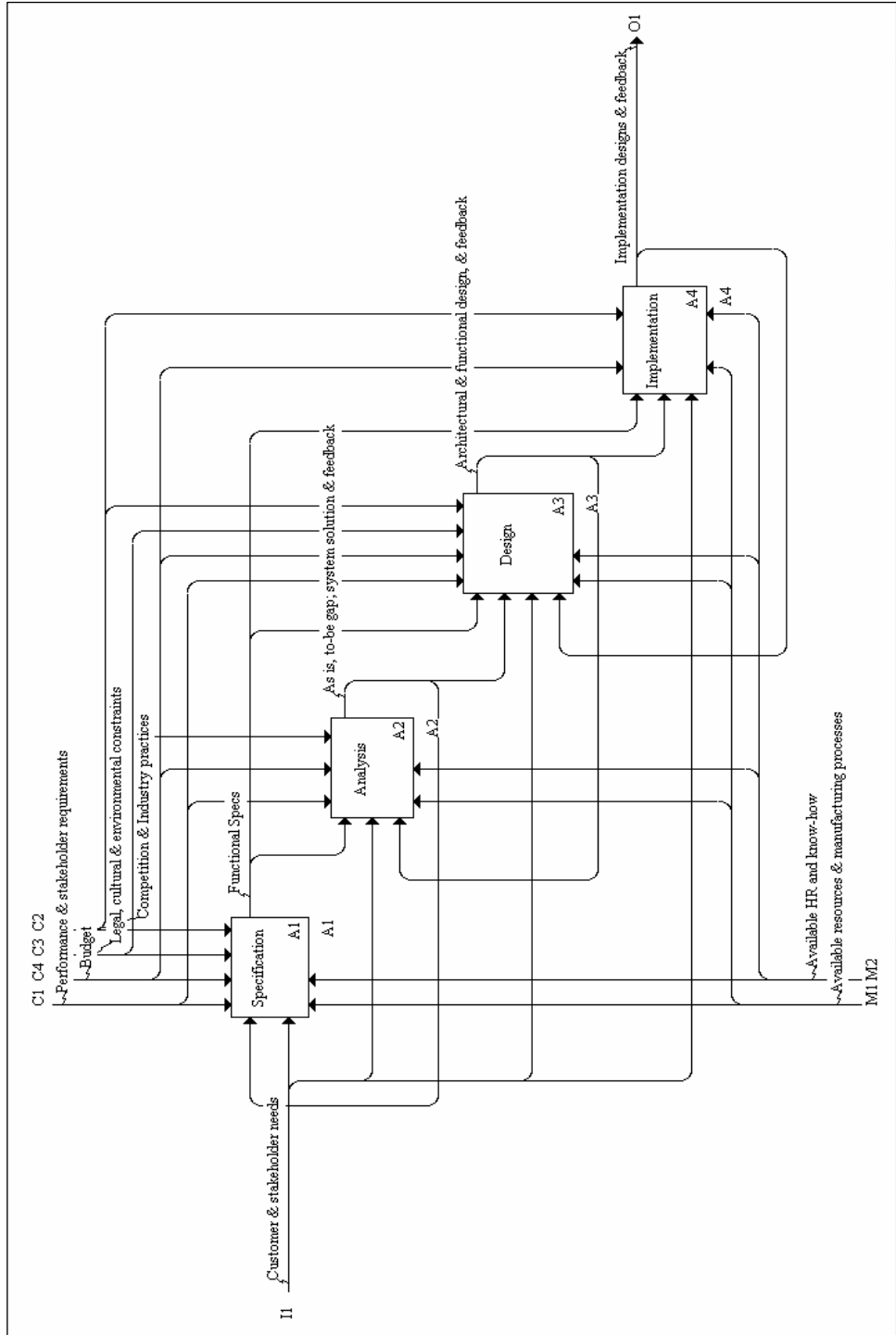
Figure 21: IDEF0 Diagram

In the context of ESE, interactions exist when there is a need for sharing passive resources, or when active resources need to collaborate or perform some work or decisions synchronously or asynchronously, or when there is a need for physical or information flow between resources. The interaction of two enterprise elements is defined as a first order interaction; interactions among three enterprise elements are defined as a second order interaction; and interactions of four enterprise elements as a third order interaction. When an enterprise element is prioritized and design choices are made for a first order interaction, design choices for lower order interactions are constrained by that prioritization. Prioritization can easily be achieved under this model by mapping class γ into a set of integers $N = \{1, 2, 3, 4\}$ in the same order as they are presented (specification=1; analysis=2; design=3, and implementation=4). Similar mapping can be done for classes α and β using the same set of integers $N = \{1, 2, 3, 4\}$. In this way, the designer is free to favor a particular prioritization of α and β (e.g. resource-based school of strategy prioritizes resources), by mapping the integer "1" to the enterprise element considered a priority, mapping the integer "2" to the next enterprise element in importance an so on.

6.3   Deliverables of the ESE Process

The outputs indicated for each activity in the IDEF0 model are the deliverables of that activity. Note that the ICOM are not single objects; on the contrary, they are entire documents, studies, or complex sets of specifications (e.g. strategy specifications, laws and regulations, industry trends, available technology, plant layout). It is out of the scope of this research to specify in detail such ICOM; although, this research does specify for what activities they are needed as input or constrains, what activities produce them as deliverables,

and the dependencies among activities via the ICOM relationships. Table 14 shows the

deliverables of the engineering activities.

Table 14: Deliverables of the ESE Process

Activity	Deliverables
Specification	Set of *functional* specifications for the whole new or modified enterprise system.
Analysis	A solution approach. A technical solution chosen among candidate alternatives based on the specifications and on the enterprise position in the industry and on a SWOT analysis to asses the internal strengths, weaknesses, opportunities, and threats to achieve the desired position in that industry
Design	Architectural and detail designs for the system, subsystems, and elements. Additionally, design includes assembly design and process design
Implementation	Detailed action plans with specific measurable objectives at all levels in the organization, and the main actions to create or transform the enterprise system. Action plans include how to promote and communicate the strategy to all the relevant parties in the organization (Molina, 2003). Implementation includes process plans, that is, how to build, assembly, and install the enterprise system

The basic criterion to assess the *quality of the deliverables* of an ESE process is compliance

with customer requirements, in this case translated into functional specifications. This is a

fundamental product development criterion, and it can be used to validate and evaluate the

quality of deliverables at any activity, sub-activity, or sub-sub-activity. Because there is a

hierarchy of specifications, they can be applied to evaluate the quality of components,

subsystems (physical, information, management), and the system as a whole.

In order to differentiate the quality of different design alternatives, Suh (2001) proposed two

axioms: the independence axiom and the information axiom. The independence axiom states

that the independence of functional requirement must be maintained. Consequently, one design is better that another if it keeps the system and subsystem functions independent from each other. The information axiom strives to minimize the information content of the design. Hence, a better design keeps the number of connections between subsystems and elements at a minimum, so that implementation is most easily accomplished (Li & Williams, 1994).

6.4   Validation of the ESE Process

The formulation of the ESE process was carried out following an inductive approach. The ESE process is a generalization supported by instances found in the literature or by empirical experience. Over two hundred references back this research, avoiding the fallacy of hasty generalization (generalization based on a too small or biased sample) (Beardsley, 1966).

Different criteria were used for validating the proposed ESE process. Three of these criteria were rigor, reliability, and validity. Rigor is guaranteed by the use of accepted methodologies (i.e. IDEF0 and product design) as the basis for the ESE process. Reliability, or the degree to which findings are independent of accidental circumstances, is guaranteed by using relevant literature, and by triangulating sources, investigators and perspectives to increase accuracy and credibility of findings (Patton, 2002). Triangulation reduces researcher bias and enhance validity (Gay & Airasian, 2000).

Other criteria was obtained from Manganelli and Hagen (2003), who after an extensive industry survey recognized that value comes from aligning the interdependent parts of the enterprise system, particularly strategy, asset portfolios, financial measures, organization, and operations. The ESE process considers all these value creating aspects and produces a

design that targets operational performance. The same authors concluded that organization structure may impede strategy implementation. The ESE framework addresses this issue by having strategy as a constraint for structure. Another criterion to increase the untapped value of existing enterprise systems is to use primary or integrated performance measures focused on the system, not on the components. The ESE framework uses a set of primary performance measures and focused on the system. Lastly, the same authors concluded that to create value it is necessary to address the enterprise components concurrently and systematically. The ESE process does exactly that.

Another validation criterion is that the ESE process is based on a validated classification scheme. Moreover, the ESE process is accompanied by a mathematical notation to ensure its complete execution. The ESE process was checked in two additional ways to further prove its validity: a) it subsumes the processes in PERA, GIM, and CIMOSA; b) it complies with the specifications set for the ESE process.

Regarding the methodologies of other enterprise architectures, PERA bases its methodology in its life cycle and in dividing the enterprise in three subsystems: manufacturing, human and organizational, and information (Li & Williams, 1994; Bernus & Nemes, 1996; Williams, 1998). The "Handbook for Master Planning and Implementation for Enterprise Integration Programs" is based on the PERA architecture (Williams et al., 1996). All the components described through the more than 300 pages of this Master Planning have a place in the proposed ESE framework; although, change management and operations management are out of the scope of this research.

As presented in the previous chapter GIM divides the enterprise system in three subsystems: information, decision, and physical. The GIM approach is focused on the decision system and uses GRAI grids to link functions with decision making. Decisions are classified by horizon of validity and period of revision. In this manner, a GRAI grid identifies and assigns functions and decisions to decision centers. It also models the flow of decisions and information between decision centers. At decision centers, GIM uses GRAI nets to model activities and decisions, their states, resources, information, and input and output objects. GRAI nets can be considered a simplified version of Petri nets, where tokens to indicate the state of the system are substituted by circles indicating activity status. GRAI nets does not include time or synchronization mechanisms (Vernadat, 1996). The use of specific methodologies, like the GRAI grid or GRAI nets, is not excluded from the ESE process; on the contrary, the ESE process allows selecting the most appropriate methodology for each activity. Regarding the scope of GIM, all the components of the methodology, decisions, decision centers, activities, resources, flows of decisions and information, inputs, and outputs, are included in the ESE process.

CIMOSA presents methodologies for function modeling, organization modeling, and information modeling. Function modeling starts by defining domains, which exchanges events and results. There are processes within each domain, triggered by events and subject to rules that constrain behavior. Functional entities perform functions. Organization modeling consists in defining a hierarchy of organization units and cells to distribute authority and responsibility. Within organization units organization cells are defined. For design specification it uses entity-relationship models and for implementation uses normalized data schemas and SQL. The components included in CIMOSA are included in

the ESE framework: functions, activities, events, resources, organization units and cells, information. CIMOSA goes beyond the scope proposed for ESE and has its own language to model information requirements and can produce models suitable for computer processing (Vernadat, 1996). In conclusion, within the scope proposed, the ESE contains the processes of PERA, GIM, and CIMOSA.

Regarding the product design perspective, the ESE process complies with the specifications set for the ESE process:

1) Product life cycle: a process model is used to describe the ESE process; life cycle is the predominant class. The engineering activities map those of product design.

2) Enterprise integration: The ESE process is based on a validated classification scheme, which is a high level model of the process. It considers the interplay or interactions between enterprise elements by the ICOM relationships. There is an ordered linking between enterprise elements and system facets through the engineering activities that realize the final product.

3) Customer requirements are translated into functional requirements and drive the overall process.

4) Strategy is a main constraint of the ESE process, it is used to create fit among enterprise elements and to guide their combination so that they reinforce one another. The resulting network of enterprise elements aims at supporting the enterprise strategy.

5) Regarding modeling principles, the ESE process has a specific purpose: producing implementation designs. The scope and domain were clearly stated by the definition of ESE while not limiting auxiliary languages or methodologies. It identifies a

viewpoint, aspects covered and left out (operations, decommission). It defines the

level of detail at each engineering activity. It considers functional decomposition by

using a hierarchy of specifications, allowing representation of abstraction levels. The

ESE process guides the building of models using a set of generic building blocks or

classes given by the enterprise elements, so they are suitable for model maintenance

and reusability.   The ESE process decouples functionality (work) from behavior

(decision), work from resources, and information from decision.

As compared to other approaches, the ESE framework represents a better model for the

engineering of an enterprise system because it covers more areas and manages interactions

among elements while offering a systematic approach and limiting the scope of the ESE

(Beardsley, 1966 Dubin, 1969).   The ESE process complies with all the requirements

imposed for validity.

CHAPTER VII

CONCLUSIONS AND FUTURE RESEARCH

This study was aimed at a better understanding of the emerging ESE field. It answered the following questions:

1) What is ESE?

2) What are its system elements and engineering activities?

3) How does ESE achieve its objectives

Specifically, the objective of this research was the development of a comprehensive framework for research in enterprise systems engineering (ESE). This framework consists of an ESE definition, an ESE classification scheme, and an ESE process. In this study, an enterprise was viewed as a system that creates value for its customers. Thus, developing the framework made use of system theory and engineering methodologies including IDEF.

ESE was defined as an engineering discipline that develops and applies systems theory and engineering techniques to specification, analysis, design, and implementation of an enterprise system for its life cycle. The proposed ESE classification scheme breaks down an enterprise system into four elements. They are work, resources, decision, and information. Each enterprise element is specified with four system facets: strategy, competency, capacity, and structure. Each element-facet combination is subject to the engineering process of specification, analysis, design, and implementation, to achieve its pre-specified performance with respect to cost, time, quality, and benefit to the enterprise.

This framework was intended for and applied to identifying research voids in the ESE discipline. It was also intended for identifying systems engineering concepts and techniques that are applicable to this emerging field. It helps harness the relationships among various enterprise aspects and bridges the gap between engineering and business practices in an enterprise. A long-term goal of this study is to establish a scientific foundation for ESE research and development.

The proposed ESE process is generic in nature. The output of an ESE effort can be a design of a partial or whole enterprise system for its physical, managerial, and/or informational layers. Thus, the proposed ESE process is applicable to a new enterprise system design or an engineering change to an existing system. To represent the ESE process, an IDEF0 model was constructed into three levels and sixty-four activities. Each activity was identified with its input, output, constraints, and mechanisms. To guide and ensure the completeness of the 64 activities in the ESE process, Petri nets were developed. A mapping between the sets of enterprise elements, system facets, and engineering activities to a set of natural numbers allows giving priority to desired enterprise elements and system facets as prioritized by the designer in reference to a particular school of thought or industry practices. The ESE process followed a product design approach, meaning that customer and stakeholder requirements are the main input. Requirements are translated into a hierarchy of functional specifications, which in turn guide the design of subsystems and elements, all sharing some responsibility for systemic performance, and keeping the enterprise system aligned with strategy.

The ESE process is underlined by the four engineering activities. It coordinates the enterprise elements, subsystems, and the system as a whole by using the set of system facets.

It guides the designer to consider the interactions among the enterprise elements and addresses the integration of physical resources, enterprise data, and application tools.

A major complication arises in the engineering of an enterprise system due to the large number of interactions among enterprise elements. Having interactions between enterprise elements means that they collaborate to achieve a common objective or there is a physical or information flow between them. Due to interactions, changes in one of the enterprise elements may affect others linked to it by the information or material flow. Following a product design approach, the ESE process provides for interaction management by the ICOM relationships in the $IDEF_0$ model and by specifications, of which both play a pivotal role.

ESE considers changes that affect the design of an enterprise system;. These changes include those that occur in the enterprise elements or in the system facets of the enterprise. However, operational changes do not change the enterprise system design and thus are not included in the ESE process. The proposed classification scheme is accompanied with the development of a *notation*, which identifies sixty-four areas of study within ESE. These areas result from the combination of enterprise elements, system facets, and engineering activities. The magnitude of the ESE fields demonstrates that ESE is an emerging research area that requires more study. Furthermore, the notation provides a means for classification and labeling of ESE activities. Thus the proposed ESE framework is an effective way to integrate all these areas of knowledge.

The merits of this research are summarized in the next section, followed by recommendations for future study, building upon the findings of this research.

7.1   Contributions

The main contribution of this research is an encompassing framework consisting of an ESE definition, a classification scheme, and an ESE process. Designing and integrating enterprise elements into a system that achieves synergy and creates value is the purpose of ESE, and an ESE framework must support such a purpose. The proposed framework does exactly that. It provides a road map for design and implementation of an integrative enterprise system. The proposed ESE framework:

1) Is generic and applicable to all industries.

2) Supports the creation and modification of an enterprise system.

3) Links various systemic aspects of the enterprise, which were usually addressed separately in the literature with little emphasis on synthesizing strategy, competency, and capacity.

4) Provides an infrastructure that integrates all areas needed to address during the engineering process of an enterprise system, unifying the approaches toward ESE.

5) Represents more areas (i.e., subsystems) of an enterprise than existing enterprise architectures do. It also allows inclusion of more elements for future extension. Thus, it overcomes a weakness in existing enterprise reference architectures, which tend to focus on one of the system (physical, managerial, or informational) layers. The proposed ESE framework places an analytical focus on enterprise elements that make up an enterprise system, and unites the three system layers mentioned above.

6) Provides a systematic approach for mapping specifications and traversing from different domains (enterprise elements) to the process that produces and installs the system, allowing alignment and opening avenues for further collaboration between

diverse areas (e.g., management, information technology, systems engineering, and industrial engineering).

7) Clears the confusion in scope and definition with a precise ESE definition and its classification scheme that serves as a generating function for consistent labeling and terminology.

8) Organizes diverse efforts in the emerging field, enabling the classification of related research efforts in enterprise systems engineering and thus signaling voids and needs for future research.

9) Serves as a basis for further development of architectures, methodologies, and (IT) tools that facilitate the engineering process of an enterprise system.

10) Provides a unique vision of the ESE field, pointing out potential capabilities of ESE in support for enterprise operations and evaluation of business partners in the process of establishing virtual enterprises.

11) Provides a means for linking the time-phased design of an enterprise system and its elements to various levels of strategy, a subject of paramount importance for today's enterprises, thus making a unique contribution.

The value of the proposed ESE framework as summarized above is the result of the convoluted value provided by each one of its components: definition, classification scheme, and process. Worth mentioning is the treatment of structure as a system facet separated from enterprise elements.

The framework addresses one goal of science, understanding, by putting forth a new theoretical foundation to create or change enterprise systems. This research was focused

fundamentally on the descriptive and qualitative side of theory building, not on hypothesis testing (Dubin, 1969). It has been demonstrated that the proposed ESE framework provides a better understanding of and approach to enterprise systems engineering in terms of its definition, scope, enterprise elements, system facets, and their interactions. Over two hundred references were cited to back the conclusions of this research, avoiding the fallacy of hasty generalization (generalization based on a too small or biased sample) (Beardsley, 1966).

It has been recognized that the value created by an enterprise comes from: 1) managing, concurrently and systematically, the interdependent parts of the enterprise system as a whole, 2) aligning resources, structure, and performance measures with strategy, and 3) using primary or integrated performance measures focused on the system, not its parts (Manganelli & Hagen, 2003). The ESE framework addresses these issues, contributing to unveiling potential value within an enterprise and to keeping aligned the enterprise elements that ultimately create value.

7.2   Recommendation for Future Research

There is much more to be done for the ESE field. As for future work, it is necessary to:

- Further decompose the ESE process, with at least one more level in the IDEF activity model.
- Refine the specification for the ICOM elements in the IDEF model; particularly those that have received little attention in the ESE field, like competencies and strategy.
- Develop an object and dynamic model for the ESE process.
- Refine the ESE process with more focus on addressing the engineering change process.

- Apply the ESE framework to (re-)designing enterprise systems and develop case studies.

- Compile ESE best practices, including change and strategy management.

- Apply and customize quantitative tools (e.g. operations research models) for various design and analysis activities in the ESE process.

- Develop generic templates, models and modules as building blocks at the enterprise's element-facet level to facilitate the ESE process in system modeling, analysis, design, implementation and integration.

- Expand the notation of the classification scheme by adding another level to the classification hierarchy. For example, resources can be readily further classified into human resource, material, equipment, and tooling, etc.

REFERENCES

Aguilar-Savén, Ruth Sara (2002a). "Analysis of Perceptions of Personnel at Organizational Levels on the Integration of Product, Functional and Process Orientation. A case study." Enterprise Inter- and Intra-Organizational Integration: Building International Consensus, The Netherlands, Kluwer Academic Publishers.

Aguilar-Savén, Ruth Sara (2002b). "Enterprise Integration into Practice". PhD Dissertation. The Department of Production Economics. Linköping Institute of Technology. Linköping, Sweden. ISBN: 91-971999-9-0.

Aguilar-Savén, Ruth Sara (2004). "Business Process Modelling. Review and Framework." International Journal of Production Economics 90 (2): 129-149.

Akao, Yoji (1990). Quality Function Deployment. Integrating Customer Requirements into Product Design. Cambridge, Massachusets Productivity Press. ISBN: 0-915299-41-0.

Appelrath, Hans-Jürgen, and Ritter, Jörg (2000). SAP R/3 Implementation. Methods and Tools. Springer. ISBN: 3-540-66863-2.

Beardsley, Monroe C. (1966). Thinking Straight. Englewood Cliffs, New Jersey Prentice-Hall, Inc. ISBN: Library of Congress Catalog Card Number 66-16388.

Berio, Giuseppe, and Vernadat, François B. (1999). "New Developments in Enterprise Modelling using CIMOSA." Computers in Industry (40): 99-114.

Bernus, Peter (2003). Organizational Design. Bernus, Nemes, Schmidt (Eds). Springer. ISBN: 3-540-00343-6.

Bernus, Peter, and Nemes, Laszlo (1996). "A Framework to Define a Gneric Enterprise Reference Architecture and Methodology." Computer Integrated Manufacturing Systems 9 (3): 179-191.

Bernus, Peter, and Nemes, Laszlo (1997). "Requirements of the Generic Enterprise Reference Architecture and Methodology." Annual Reviews in Control 21: 125-136.

Bernus, Peter, and Nemes, Laszlo (2003). <u>Handbook on Enterprise Architecture.</u> Bernus, Nemes, Schmidt (Eds). Springer. ISBN: 3-540-00343-6.

Bruce, Thomas A. (1992). <u>Designing Quality Databases with IDEF1X Information Models.</u> Dorset House. ISBN: 0-932633-18-8.

Chakrabarti, Kisor Kumar (1995). <u>Definition and Induction. A Historical and Comparative Study.</u> Society for Asian and Comparative Philosophy. University of Hawai's Press. ISBN: 0-8248-1658-7.

Chandra, Charu, and Kumar, Sameer (2001). "Enterprise Architectural Framework for Supply-chain Integration." <u>Industrial Management & Data Systems</u> 101 (6): 290-303.

Checkland, Peter (1982). <u>Systems Thinking, Systems Practice.</u> John Willey & Sons. ISBN: 0-471-27911-0.

Checkland, Peter, and Holwell, Sue (1998). <u>Information, Systems and Information Systems.</u> Lancaster University, UK John Willey & Sons. ISBN: 0-471-95820-4.

Checkland, Peter, and Scholes, Jim (1990). <u>Soft Systems Methodology in Action.</u> John Wiley & Sons. ISBN: 0-471-92768-6.

Chen, David, Vallespir, Bruno, and Doumeingts, Guy (1997). "GRAI Integrated Methodology and its Mapping onto Generic Enterprise Reference Architecture and Methodology." <u>Computers in Industry</u> (33): 387-394.

Chen, David, Vallespir, Bruno, and Doumeingts, Guy (2003). "Preliminary Design: Translating Requirements to Design Specifications". <u>Handbook of Enterprise Architecture</u>: Bernus, Nemes, Schmidt (Eds). Springer.

CIMOSA (2003). CIMOSA Specification. Electronic Source: http://www.cimosa.de/About_us.htm. CIMOSA Association.

Collis, D. J. (1994). "Research Note: How Valuable are Organizational Capabilities?" <u>Strategic Management Journal</u> 15: 143-152.

Copi, Irving M. (1982). <u>Introduction to Logic.</u> Sixth Edition. Macmillian Publishing Co., Inc. ISBN: 0-02-324920-x.

Copi, Irving M., and Burguess-Jackson, Keith (1995). <u>Informal Logic.</u> Prentice Hall. ISBN: 0-13-229048-0.

Coulter, Mary (2002). <u>Strategic Management in Action.</u> ISBN: 0-13-040006-8.

Dangayach, G.S., and Deshmukh, S.G. (2001). "Manufacturing Strategy. Literature Review and Some Issues." <u>International Journal of Operations and Production Management</u> 21 (7): 884-932.

Davenport, Thomas H., and Short, James E. (1990). The New Industrial Engineering: Information Technology and Business Process Redesign. <u>Sloan Management Review</u>**: 11-27.

David, Fred R. (1997). <u>Strategic Management.</u> Prentice-Hall, Inc. ISBN: 13-486011-X.

Drucker, Peter F. (1999). <u>Management Challenges for the 21st Century.</u> Harper Business. ISBN: 0-88730-998-4.

Dubin, Robert (1969). <u>Theory Building.</u> New York The Free Press. ISBN: Libray of Congress Catalog Card Number: 69-10480.

Frankl, Paolo, and Rubik, Frieder (2000). <u>Life Cycle Assestment in Industry and Business. Adoption Patterns, Applications and Implications.</u> Springer-Verlag Berlin Hidelberg. ISBN: 3-540-66469-6.

Frohlich, Markham T., and Dixon, J. Robb (2001). "A Taxonomy of Manufacturing Strategies Revisited." <u>Journal of Operations Management</u> (19): 541-558.

Gaither, Norman, and Fraizer, Greg (1999). <u>Production and Operations Management.</u> South-Western College Publishing. ISBN: 0-538-89108-4.

Gay, Lorraine Rumbel, and Airasian, Peter (2000). <u>Educational Research. Competencies for Analysis and Application.</u> Sixth Edition. Merrill - Prentice Hall. ISBN: 0-13-096103-5.

Georgakopoulos, Dimitrios, Hornick, Mark, and Sheth, Amit (1995). "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure." <u>Distributed and Parallel Databases</u> 3: 119-153.

Giachetti, Ronald E. (2004). "A Framework to Review the Information Integration of the Enterprise." <u>International Journal of Production Research</u> 42 (6): 1147-1166.

Giachetti, Ronald E., Martinez, Luis D., Sáenz, Oscar A., and Chen, Chin-Sheng (2003). "Analysis of the Structural Measures of Flexibility and Agility using a Measurement Theoretical Framework." <u>International Journal of Production Economics</u> 86 (1): 47-62.

Goold, Michael, and Campbell, Andrew (2002). <u>Designing Effective Organizations: How to Create Structured Networks.</u> San Francisco Jossey-Bass. ISBN: 0787960640.

Goranson, Ted (2003). "Capability Improvement". <u>Handbook for Enterprise Architecture</u>: Bernus, Nemes, Schmidt (Eds). Springer.

Grunninger, Michael (2003). "Enterprise Modelling". <u>Handbook of Enterprise Architecture</u>: Bernus, Nemes, Schmidt (Eds). Springer.

Hanson, Barbara Gail (1995). <u>General Systems Theory. Beginning with Wholes.</u> Taylor & Francis. ISBN: 1-56032-345-0.

Hempel, Carl G. (1965). <u>Aspects of Scientific Explanation and other Essays in the Philosophy of Science.</u> New York The Free Press. ISBN: Library of Congress Catalog Card Number 65-15441.

Hillier, Frederick S., and Liberman, Gerald J. (2002). <u>Introduction to Operations Research.</u> McGraw-Hill. Seventh Edition. ISBN: 0-07-246121-7.

IFIP-IFAC, Task Force on Architectures for Enterprise Integration (2003). "Handbook of Enterprise Architecture". Bernus, Peter, Blazewicz, Jacek, Schmidt, Günter and Shaw, Michael: Springer-Verlag.

ISEE (2003). International Society for Enterprise Engineering. Electronic source: http://www.hissight.com/iseenet/glossary.htm#sectA.

ISO (1999a). International Organization for Standardization. Concepts and Rules for Enterprise Models. International Standard 14258.

ISO (1999b). <u>International Organization for Standardization. Requirements for Enterprise Reference Architectures and Methodologies. International Standard 15704</u>. NIST.

Jayachandra, Y. (1994). Re-Engineering the Networked Enterprise. McGraw-Hill, Inc. ISBN: 0-07-032017-9.

Jin, Chun (1999). "Modeling and Performance Evaluation of Complex Traffic Control Systems Using Stochastic Timed Petri Nets". Master thesis. Computer Science Department. Florida International University. Miami.

Kalpic, Brane, Pandza, Krsto, and Bernus, Peter, Eds. (2003). Strategy as a Creation of Corporate Future. Handbook of Enterprise Architecture: Bernus, Nemes, Schmidt (Eds). Springer.

Kaplan, Robert, and Norton, David (2000). The Strategy Focused Organization: How Balanced Scorecard Organizations Thrive in the New Business Environment. Harvard Business School Publishing Corporation. ISBN: 1-57851-250-6.

Kaplan, Robert S., and Norton, David P. (1996). Translating Strategy into Action. The Balanced Scorecard. Harvard Business School Press. ISBN: 0-87584-651-3.

Kaplan, Robert S., and Norton, David P. (2004). Strategy Maps. Converting Intangible Assets into Tangible Outcomes. Hardvard Business School Press. ISBN: 1-591-39-134-2.

KBSI (2003). A Structured Approach to Enterprise Modeling and Analysis. Retrieved from Knowledge Based Systems website: http://www.idef.com/default.html.

Keidel, Robert (1995). Seeing Organizational Patterns: A new theory and language of organizational design. Berrett-Koehler Publishers. ISBN: 1881052656.

Kettinger, William J., and Teng, James T.C. (1998). "Aligning BPR to Strategy: a Framework for Analysis." Long Range Planning 31 (1): 93-107.

Kim, Cheol-Han, Weston, Richard H., Hodgson, Allan, and Lee, Kyung-Huy (2003). "The complementary use of IDEF and UML modelling approaches." Computers in Industry 50: 35-56.

Kim, Kwang Jae, and Moskowitz, Herbert (1997). "Quality Function Deployment: Optimizing Product Designs." Integrated Product, Process and Enterprise Design. Ben Wang (Eds). Chapman & Hall. London.

Kosanke, Kurt (1995). "CIMOSA - Overview and status." Computers in Industry (27): 101-109.

Kosanke, Kurt (1999). "Enterprise Engineering and Integration - Towards Real-time Decision Support". Interkama-ISA-Tech Conference, Düsseldorf, Germany.

Kosanke, Kurt, Nell, J.G., Vernadat, François, and Zelm, Martin (1998). "Enterprise Integration - International Consensus (EI - IC)". EP21859. Proc. Integration in Manufacturing, ISBN: 90-5199-426-5.

Kosanke, Kurt, and Nell, James G. (1999a). "Enterprise Organisations and the New Enterprise Paradigms". Proceedings of the 14th. IFAC World Congress, Beijing, P.R. China.

Kosanke, Kurt, and Nell, James G. (1999b). "Standardisation in ISO for enterprise engineering and integration." Computers in Industry (40): 311-319.

Kosanke, Kurt, Vernadat, F., and Zelm, M. (1999). "CIMOSA: Enterprise Engineering and Integration." Computers in Industry (40): 83-97.

Kosanke, Kurt, and Zelm, Martin (1999). "CIMOSA Modelling Process." Computers in Industry (40): 141-153.

Kotler, Philip (1994). Marketing Management. Analysis, Planning, Implementation, and Control. Prentice-Hall. ISBN: 0-13-722851-1.

Kotler, Philip, and Armstrong, Gary (2001). Principles of Marketing. Prentice hall. ISBN: 0-13-064853-1.

Li, Hong, and Williams, Theodore J. (1994). A Formalization and Extension of the Purdue Enterprise Reference Architecture and the Purdue Methodology. Technical Report. Purdue Laboratory for Applied Industrial Control. Purdue University. West Lafayette, Indiana.

Lim, Soon Huat, Juster, Neal, and Pennington, Alan de (1997). "Enterprise Modelling and Integration: A Taxonomy of Seven Key Aspects." Computers in Industry (34): 339-359.

Lummus, Rhonda R., Krumwiede, Dennis W., and Vokurka, Robert J. (2001). "The Relationships of Logistics to Supply Chain Management: Developing a Common Industry Definition." Industrial Management & Data Systems 101 (8): 426-431.

Luo, Wenhong, and Tung, Y. Alex (1999). "A Framework for Selecting Business Process Modeling Methods." Industrial Management & Data Systems 99 (7): 312-319.

Madu, Christian N. (2000). House of Quality (QFD) in a Minute. Fairfield, CT. Chi Publishers. ISBN: 0-9676023-0-0.

Malone, T.W., and Crowston, K. (1994). "The interdisciplinary study of coordination." ACM Computing Surveys 26 (1): 87-119.

Manganelli, Raymond L., and Hagen, Brian W. (2003). Solving the Corporate Value Enigma. Amacon. ISBN: 0-81440692-0.

Martin, James (1995). The Great Transition. Using the Seven Disciplines of Enterprise Engineering to Align People, Technology, and Strategy. Amacon. ISBN: 0-8144-0315-8 (pbk).

Merriam-Webster (2004). OnLine Dictionary. http://www.webster.com. Access date: 03-05-2004 to 07-04-2005.

Miles, Matthew B., and Huberman, A. Michael (1994). Qualitative Data Analysis. An Expanded Sourcebook. Second Edition. Sage Publications. ISBN: 0-8039-4653-8.

Miller, Jeffrey G., and Roth, Aleda V. (1994). "A Taxonomy of Manufacturing Strategies." Management Science 40 (3): 285-302.

Miller, Thomas E., and Berger, Daryle W. (2001). Totally Integrated Enterprises. A Framework and Methodology for Business and Technology Improvement. Raytheon Professional Services LLC. ISBN: 1-57444-303-8.

Mintzberg, Henry (1979). The Structuring of Organizations. Englewood Cliffs, N.J Prentice-Hall. ISBN: 0138552703.

Mintzberg, Henry (1992). Structure in Fives: Designing Effective Organizations. Englewood Clifts, N.J Prentice Hall. ISBN: 013855479X.

Molina, Arturo (2003). "Developing the Enterprise Concept - The Business Plan". Handbook for Enterprise Architecture: Bernus, Nemes, Schmidt (Eds). Springer.

Montgomery, Douglas C., Runger, George C., and Hubele, Norma F. (2001). Engineering Statistics. Second Edition. John Wiley & Sons, Inc. ISBN: 0-471-38879-3.

Mosey, Simon P., Wookcock, David J., and Clare, Jeremy N. (2003). "Process to Support Strategic Ambition in SMEs." Production Planning & Control 14 (4): 372-383.

Mukherji, Ananda (2002). "The Evolution of Iformation Systems: Their Impact on Organization and Structures." Management Decision 40 (5): 497-507.

Nalbone, John, Vizdos, Michael, and Ambler, Scott (2004). Adopting the Enterprise Unified Process. Ronin International, Inc.

Nell, James G. (1999). Enterprise Representation: A different Paradigm for Designing Process-Interoperability Standards. Unpublished. National Institute of Standards and Technology (NIST). Gaithersburg, MD.

Nell, James G. (2000). Designing Standards to Improve Process Interoperability. National Institute of Standards and Technology (NIST). Dowloaded July 22-2003 http://www.mel.nist.gov/sc5wg1/wg1vision.htm.

NIST (1993a). Integration Definition for Function Modeling (IDEF0). National Institute of Standards and Technology. Processing Standards Publication 183.

NIST (1993b). Integration Definition for Information Modeling (IDEF1X). National Institute of Standards and Technology, USA. Federal Information Processing Standards Publication 184.

Noran, Ovidiu (2003). "An Analysis of the Zachman Framework for Enterprise Architecture from the GERAM Perspective." Annual Reviews in Control 27 (2): 163-183.

Olegario, Cielito, and Bernus, Peter (2003). "Modelling the Management System - Enterprise Management and Activities". Handbook of Enterprise Architecture: Bernus, Nemes, Schmidt (Eds). Springer.

OMB (2004). OMB Enterprise Architecture. Assessment v1.0 Guidelines. Office of Management and Budget. Federal Enterprise Architecture (FEA) Program Management Office.

Ortiz, Angel, Lario, Francisco, and Ros, Lorenzo (1999). "Enterprise Integration - Business Process Integrated Management: A Proposal for a Methodology to Develop Enterprise Integration Programs." Computers in Industry (40): 155-171.

Patton, Michael Quinn (2002). Qualitative Research & Evaluation Methods. Sage. ISBN: 0-7619-1971-6.

Pearlson, Keri E. (2001). Managing and Using Information Systems. John Wiley & Sons, Inc. ISBN: 0-471-32001-3.

Pinedo, Michael (2001). Scheduling. Theory, Algorithms, and Systems. Prentice Hall, Inc. ISBN: 0-13-028138-7.

Porter, Michael E. (1985). Competitive Advantage: Creating and Sustaining Superior Performance. New York: Free Press. ISBN: 0029250900.

Porter, Michael E. (1996). "What is Strategy?" Harvard Business Review (November-December): November-December, 61-78.

Presley, Adrien, and Liles, Donald (1996). "Enterprise Modeling within an Enterprise Engineering Framework". Winter Simulation Conference, San Diego.

Presley, Adrien, Sarkis, Joseph, Barnett, William, and Liles, Donald (2001). "Engineering the Virtual Enterprise: An Architecture-Driven Modeling Approach." The International Journal of Flexible Manufacturing Systems (13): 145-162.

Ravindran, A., Phillips, Don T., and Solberg, James J. (1987). Operations Research. Principles and Practice. John Wiley & Sons. Second Edition. ISBN: 0-471-08608-8.

Reithofer, Walter, and Naeger, Georg (1997). "Bottom-up Planning Approaches in Enterprise Modelling - The Need and the State of the Art." Computers in Industry (33): 223-235.

Robinson, Richard (1968). "Definition": Oxford University Press, Ely House, London.

Rodriguez, Adolfo, Valle, Flor de Maria, and Chaix, Yves (2004). Marco Conceptual de un Gobierno Electrónico para Nicaragua. Resumen Técnico (Conceptual Framework for the Nicaraguan E-Goverment. Technical Summary). Public Sector Reform Coordination Unit. Managua, Nicaragua.

Rowe, Frantz, Truex, Duane, and Kvasny, Lynette (2004). "Cores and Definitions: Building the Cognitive Legitimacy of the Information Systems Discipline across the Atlantic". Relevant Theory and Informed Practice: looking forward from a 20 year perspective on IS research: B. Kaplan, D. Truex, D. Wastell and A.T. Wood-Harper (Ed.), Kluwer, Boston**:** 83-102.

Sackett, Peter J., Maxwell, Douglas J., and Lowenthal, Paul A. (1997). "Customizing Manufacturing Strategy." Integrated Manufacturing Systems 8 (6): 359-364.

Saenz, Oscar A., and Chen, Chin-Sheng (2004). "Towards a Framework for Enterprise Systems Engineering". Second LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2004). Challenges and Opportunities for Engineering Education, Research and Development. June 2-4, 2004, Miami, Florida, USA.

Scheer, August-Wilhelm (1998). ARIS - Business Process Frameworks. Springer-Verlag. ISBN: 3-540-64439-3.

Scheer, August-Wilhelm (1999). ARIS - Business Process Modeling. Springer-Verlag. ISBN: 3-540-64438-5.

Simchi-Levi, David, Kaminsky, Phillip, and Simchi-Levi, Edith (2000). Designing and Managing the Supply Chain. Irwin / McGrawn-Hill. ISBN: 0-07-028594-2.

Smith, Thomas M., and Reece, James S. (1999). "The Relationship of Strategy, Fit, Productivity, and Business Performance in a Service Setting." Journal of Operations Management (17): 145-161.

Sternemann, K.H., and Zelm, Martin (1998). "Enterprise Modeling for Operational Decision Support". Proceedings IEEE SMC'98 Internationa Conference on Systems, Manufacturing and Cybernetics, October, 1998. San Diego, USA.

Suh, Nam Pyo (2001). Axiomatic Design: Advances and Applications. Oxford University Press. ISBN: 0-19-513466-4.

Thomé, Bernhard (1993). "Systems Engineering. Principles and Practice of Computer-based Systems Engineering." Thomé, Bernhard (Ed): John Willey & Sons.

Tolle, Martin, and Vesterager, Johan (2003). "Developing the Business Model - A Methodology for Virtual Enterprises". Handbook for Enterprise Architecture: Bernus, Nemes, Schmidt (Eds). Springer.

Toulmin, Stephen, Rieke, Richard, and Janik, Allan (1979). An Introduction to Reasoning. Macmillian Publishing Co., Inc. ISBN: 0-02-421030-7.

Truex, Duane P., Baskerville, Richard, and Klein, Heinz (1999). "Growing Systems in Emergent Organizations." Communications of the ACM 42 (8): 117-123.

Ulrich, Karl T., and Eppinger, Steven D. (2000). Product Design and Development. Irwin, MacGraw-Hill. ISBN: 0-07-229647-x.

Upplington, Greg, and Bernus, Peter (2003). "Analysing the Enterprise Situation and Refining Strategy". Handbook of Enterprise Architecture: Bernus, Nemes, Schmidt (Eds). Springer.

van der Aalst, Wil M. P. (2002). "Making Work Flow: On the Application of Petri Nets to Business Process Management". Application and Theory of Petri Nets 2002. 23rd International Conference, ICATPN 2002, Adelaide, Australia, Springer.

Veasey, Philip W. (2001). "Use of Enterprise Architectures in Managing Strategic Change." Business Process Management Journal 7 (5): 420-436.

Vernadat, Francois B. (1996). Enterprise Modeling and Integration: Principles and Applications. UK Chapman & Hall. ISBN: 0-412-60550-3.

Vernadat, Francois B. (2001). "UEML: Towards a Unified Enterprise Modeling Language". 3rd. Conference Francophone de Modélisation et Simulation "Conception, Analyse et Gestion des Systèmes Industriales" MOSIM'01, Troyes, France.

Vernadat, Francois B. (2002). "Enterprise Modeling and Integration (EMI): Current Status and Research Perspectives." Annual Reviews in Control (26): 15-25.

Verville, Jacques, and Halingten, Alannah (2002). "An Investigation of the Decision Process for Selecting an ERP Software: The Case of ESC." Management Decision 40 (3): 206-216.

Vogler, Walter (1992). <u>Modular Construction and Partial Order Semantics of Petri Nets.</u> Lecture
    Notes in Computer Science. Springer-Verlag. ISBN: 0-387-55767-9.

Watkins, Harry S. (1997). <u>Integrated Product, Process and Enterprise Design.</u> Ben Wang (Ed).
    Chapman & Hall. ISBN: 0-412-62020-0.

Whitman, Larry, Ramachandran, Kartik, and Ketkar, Vikram (2001). "A Taxonomy of a Living
    Model of the Enterprise". <u>Proceedings of the 2001 Winter Simulation Conference</u>.

Whitten, Jeffrey L., Bentley, Lonnie D., and Dittman, Kevin C. (2001). <u>Systems Analysis and
    Design Methods.</u> McGraw-Hill, Irvin. ISBN: 0-07-231539-3.

Williams, Theodore J. (1998). The Purdue Enterprise Reference Architecture and Methodology
    (PERA). Handbook of Life Cycle Engineering: Concepts, Tools and Techniques; Edited
    by A. Molina, J.M. Sanchez, A. Kusiak; Published by: Chapman & Hall, London.

Williams, Theodore J. (1999). PERA and GERAM: Establishment of the Place of the Human in
    Enterprise Integration. IFAC Congress, Beijing, July. Institute for Interdisciplinary
    Engineering Studies, Purdue University.

Williams, Theodore J., and Li, Hong (1998). PERA and GERAM--Enterprise Reference
    Architectures in Enterprise Integration. DIISM. Kluwer Academic Publishers. Institute
    for Interdisciplinary Engineering Studies, Purdue University.

Williams, Theodore J., Li, Hong, Bernus, Peter, Uppington, Gregory, and Nemes, Laszlo (1998).
    "The Life Cycle of an Enterprise". <u>Handbook of Life Cycle Engineering: Concepts,
    Tools and Techniques</u>; Edited by A. Molina, J.M. Sanchez, A. Kusiak. London,
    Chapman & Hall.

Williams, Theodore J., Rathwell, Gary A., and Li, Hong (1996). <u>A Handbook on Master
    Planning and Implementation for Enteprise Integration Programs. Based On The Purdue
    Enterprise Reference Architecture and the Purdue Methodology. Report Number 160</u>.
    Institute for Interdisciplinary Engineering Studies. Purdue University. West Lafayette,
    Indiana 47907.

Wilson, Brian (1984). <u>Systems: Concepts, Methodologies, and Applications.</u> John Willey &
    Sons. ISBN: 0-471-90443-0.

Wu, Bin, and Ellis, Ray (2000). "Manufacturing strategy analysis and manufacturing information system desing: Process and application." <u>International Journal of Production Economics</u> (65): 55-72.

Xia, F. (1999). "Look Before you Leap: On Some Fundamental Issues in Software Engineering Research." <u>Information and Software Technology</u> (41): 661-672.

Zachman, John A. (1999). "A Framework for Information Systems Architecture." <u>IBM Systems Journal</u> 38 (2&3): 454-470.

Zachman, John A. (2003). <u>The Zachman Framework: A Primer for Enterprise Engineering and Manufacturing</u> (electronic book). Excerpts from "The Zachman Framework for Enterprise Architecture". Zachman International. 07-22-2003.

Zelm, Martin (2003). <u>Resource Requirements of Enterprise Management.</u> Bernus, Nemes, Schmidt (Eds). Springer. ISBN: 3-540-00343-6.

Zelm, Martin, and Kosanke, Kurt (1999). "Enterprise Modeling - Towards a User Oriented Methodology". <u>Proceedings Intelligent Industrial Automation (IIA'99)</u>, Jun 02-04, Genoa, Italy.

Zelm, Martin, and Kosanke, Kurt (2001). "A Modeling Language for User Oriented Enterprise Modeling". <u>3rd. Conference Francophone de Modélisation et Simulation "Conception, Analyse et Gestion des Systèmes Industriales" MOSIM'01</u>, April 25-27, Troyes, France.

Zülch, Gert, Rinn, Andreas, and Strate, Oliver (2001). "Dynamic analysis of changes in decisional structures of production systems." <u>Int. J. Production Economics</u> (69): 239-252.

Zwegers, Arian J.R., and Gransier, Theo A.G. (1995). "Managing Re-Engineering with the CIMOSA Architectural Framework." <u>Computers in Industry</u> (27): 143-153.

APPENDIX

IDEF0 Model for ESE

A0: ENGINEERING AN ENTERPRISE SYSTEM

From a product design stand point, customer requirements are translated into functional specifications (Ulrich & Eppinger, 2000; Suh, 2001). The product design approach allows not only for the traceability of specifications, and to link every final solution to a customer or stakeholder requirement, but also allows the following up of the impact of changes in one enterprise element on other interacting enterprise elements.

The ESE process has the following general inputs, constraints, and mechanisms:

- General Input: Customer & stakeholder needs.

- General constraints: legal, cultural & environmental constraints; competition & industry practices; performance & stakeholder requirements; available budget.

- General mechanisms: Available manufacturing processes, technology, know-how, and resources.

Node A0 in the IDEF0 model shows the top level of the ESE process. Node A0 clearly shows that engineering activities are the heart of the ESE process. Tables are used to further describe the activities.

Table A0: Level 1 Activities in the IDEF0 Diagram

Activity	Description
A1: Specification	The output the specification activity is a set of functional specifications. These serve as input to analysis. They are also considered input to design and implementation as a way to continuously check that the technical solution satisfies the original specifications. There are 16 sets of specifications, which are the pair combination of enterprise elements and system facets.
A2: Analysis	The inputs of the analysis activity are the enterprise specifications. The output is the as-is state, a technical solution approach, or the to-be state, and the as-is/to-be gap. Analysis focuses on system level solutions and its possible general configuration without considering available components from the market, constraining the universe of final design solutions.
A3: Design	Design has the functional specifications as input. Design translates the functional specifications and the technical solution approach into design parameters. The output of design is an architectural design, decomposed at lower levels until defining subsystems, enterprise elements and their interactions, their capabilities, capacities, and structure.
A4: Implementation	The inputs of implementation are the functional specs and the architectural and functional design. The output of implementation are implementation plans, which are the equivalent of a process plan, the one that will deliver the enterprise system, an assembly plan, the one that specifies how to integrate the system, and a deployment plan, the one that establish how to install the system and train users.

Table A1: Levels 2 and 3 of the Specification Activity

Activity	Sub-activity	Description
A11: Strategy specification	A111: Work strategy specification	The output of strategy specification is the definition of the work type that will make up the core business processes. Core business processes achieve the proposed objectives and produce core products. These specifications define work policies and the level of work specialization (e.g. focused vs. diversified product) required to achieve performance targets of cost, lead time, and quality.
	A112: Resource strategy specification	The outputs of resource strategy specification are of two kinds: one financial and one technological. Financially, it implies the identification of the necessary financial resources (global needs) and the intended capital structure of the enterprise, which in turn depends on the expected amounts of financial resources provided by the stockholders and other sources (suppliers, banks, and other creditors). The preliminary allocation of financial and other resources according to the required capacity and capabilities are also specified. Technologically, it states initial considerations of technological resources, how to get them, identification of potential supply chain relationships (raw material sources and distribution channels). The resource strategy specs include, in general terms, the extent of automation (Williams, 1998).
	A113: Decision strategy specification	The output of decision strategy specification involves the identification of the highest level of decision frameworks within the enterprise; that is, the enterprise objectives, constraints, and timeframes that will be passed down to the lower levels in the enterprise structure. These specifications are the product of rational decision-making; that is, there is close relationship between the ends and the means to achieve those ends (Frankl & Rubik, 2000).
	A114: Information strategy specification	The output of information strategy specification establishes the role of the future enterprise's information system in terms of providing support to implement the enterprise strategy (Pearlson, 2001). In terms of the Zachman's framework, information strategy specification corresponds to the system scope from the perspective of the planner, defining the important objects (data) to manage (including performance), the core business processes or work, major organizational units to support, the location or network where the enterprise will operate, the timeframes, and the goals of the future information system (Zachman, 2003). These specifications include an initial plan to gather user requirements and considerations for in-house development vs. acquisition of information and know-how.

Table A1: Levels 2 and 3 of the Specification Activity

Activity	Sub-activity	Description
A12: Competency specification	A121: Work competency specification	The outputs of work competency specification are specifications stating what the system needs to do in order to satisfy customers' and stakeholders' requirements. An enterprise system can be seen as a network of competency units. Work competency specifications establish what to do within an enterprise system and what work will be done outside its boundaries. It defines a make and outsource plan. These specifications also define workflows to accomplish a business process (Georgakopoulos et al., 1995).
	A122: Resource competency specification	Resources competency specification uses the make and buy (outsource) plan as input to generate its outputs: the types of resources needed and the suppliers, main resources and locations, and the system coordination needs. The resources competency specifications establish the type of resources needed, and the expected participants in the supply chain (identification of business partners) that will provide the capabilities needed. The main types of resources (HR, manufacturing technology, and IT) and its distribution (geographical location) are identified in order to set up potential flows of crews, raw materials, final products, and other resources. Together with the work competency specifications, these specifications form the value chain strategy, the integration (vertical or horizontal) level and collaboration links with a supply chain (Molina, 2003).
	A123: Decision competency specification	The output of decision competency specification defines the needs for competency units and the type of expected relationships between the main resources (Bernus, 2003), which in turn define how decisions, objectives, constraints, and timeframes flow through lower levels in the enterprise system.
	A124: Information competency specification	Information competency specifications use as inputs the outputs of work competency specs, resource competency specs, and decision competency specs, to generate its outputs: • It translates customer requirements into functional specifications for the information system, that is, the information required to perform work or decision making. • The network model, the logical model based on the locations to serve, the distribution of resources (geographical layout), the planned supply chain strategy, and the coordination needs among resources. • Information flows and main events between subsystems. Information competency is about making information and knowledge available to the one that needs it in an enterprise. This has been called information capital by Kaplan and Norton (2004).
A13: Capacity specification	A131: Work capacity specification	The output of work capacity specification is an order of magnitude of the work required to be done by the system and automation level.

Table A1: Levels 2 and 3 of the Specification Activity

Activity	Sub-activity	Description
A13: Capacity specification	A132: Resource capacity specification	The output of resource capacity specification is an order of magnitude of resources required by the system to perform productive and managerial activities. This allows, as mentioned by Zelm (2003), the estimation of resources' investments and recurrent cost.
	A133: Decision capacity specification	The inputs of decision capacity specification are work and resource capacities. The output of decision capacity specifications is an order of magnitude and types of decisions to be made at system level.
	A134: Information capacity specification	The inputs of information capacity specification are decision capacity, resources capacity, and work capacity. The output of information capacity specifications is an order of magnitude of information required to store, process, or transmit, in order to perform work and decisions, and support resources, at system level.
A14: Structure specification	A141: Work structure specification	The output of work structure specification is the definition of a criterion to aggregate the work element (e.g. knowledge or functional specialization, geography, products, technology) that will guide the design of tasks, activities and business processes.
	A142: Resource structure specification	The output of resource structure specification is a criterion to aggregate resources. Resources are the most important component of the enterprise structure. They will constrain the main functionality and behavior of the enterprise system. The resulting properties of the enterprise system will emerge as a result of the structure (or aggregation) of resources (Chen et al., 2003; Bernus, 2003).
	A143: Decision structure specification	The output of decision structure specification is a criterion to aggregate decisions; a set of core enterprise decisions organized by their horizon of validity and their period of revision (e.g. GRAI-Grid) (Vernadat, 1996; Olegario & Bernus, 2003). A basic design criterion is to minimize dependency among decisions; that is, identify the interactions among decisions, then identify independent groups of decisions, and finally, regroup decisions to reduce dependency between them (Chen et al., 2003).
	A144: Information structure specification	The output of information structure specification is a criterion to aggregate information; the main classes of data are established; it creates a semantic model with the business entities and their relationships (Zachman, 2003).

Table A2: Levels 2 and 3 of the Analysis Activity

Activity	Sub-activity	Description
A21: Strategy analysis	A211: Work strategy analysis	The output of work strategy analysis is the evaluation of the alignment between existing or projected work and the planned enterprise mission, vision, and objectives.
	A212: Resource strategy analysis	The output of resource strategy analysis is the evaluation of resources (alignment, availability and use of financial, human, and technological resources and knowledge) to support the strategy specs.
	A213: Decision strategy analysis	The output of decision strategy analysis is an assessment of aspects that may influence the main decisions outlined in the enterprise specs. It refers to the assessment and current state of target markets (customers, suppliers, competitors, and products), economy and business environment conditions, technology and industry trends, government, and legal aspects. The Porter's five forces analysis (suppliers, customers, industry competition, new entrants, and substitute products) can be used during this activity.
	A214: Information strategy analysis	The output of information strategy analysis is the current and desired level of support that information technology (IT) and information systems (IS) will provide for the achievement of the enterprise objectives. The information strategy analysis assesses the potential use of information systems and information technology in the business, the risks associated with an eventual investment in IT/IS (sustainability, ROI, change management requirements, what does competitors do), the identification of IT/IS stakeholders (potential internal and external users, legal framework), and project management level of maturity. During this activity a plan for gathering information systems user requirements is made, together with non functional requirements that will be expected from the IT infrastructure and related services (web services, network services).
A22: Competency analysis	A221: Work competency analysis	The output of work competency analysis is the set of work specifications, which are decomposed at subsystem level. Feasibility of the work competence specs in term of their customer or management orientation, and how they contribute with the desired enterprise performance (cost, time, quality, or benefit) and other objectives. Work competency analysis proposes work types to satisfy specs and fill the gap between the as-is and to-be system.

Table A2: Levels 2 and 3 of the Analysis Activity

Activity	Sub-activity	Description
A22: Competency analysis	A222: Resource competency analysis	The output of resource competency analysis is the set of resource specs decomposed at subsystem level and the identification of the types of resources to satisfy specs. Core competencies of resources are identified for existing enterprise systems. Resource competency analysis evaluates the adequacy of current core capabilities and competencies to support the desired state of the enterprise system and any new product to offer. For new enterprise systems, resource competency analysis identifies types of resources required for the business opportunity (Molina, 2003). Active and passive resources may flow. Resources competency analysis provides information to evaluate the feasibility of resources flow, e.g. transshipment nodes, destinations, restrictions, and costs in the network (locations and linkages) of possible flows.
	A223: Decision competency analysis	The output of decision competency analysis is the set of decision specs decomposed at subsystem level, and the advantages and disadvantages of the potential decision channels between and within potential decision centers, and mainly between their active resources.
	A224: Information competency analysis	Using work competency and resource competency analysis as inputs, information competency analysis outputs are the functional specifications decomposed at subsystem level. During this activity the selection of a methodology for information system development is made, as the Rational Unified Process (Rodriguez et al., 2004) or Zachman's and its variations (Whitten et al., 2001), which consider the eliciting and gathering of user requirements. Data modeling, process modeling, and use case diagrams are used to document this activity. Information competency analysis evaluates the feasibility of the planned information channels and information flows. This activity elicits the problems and opportunities associated with the potential information flows, and the work and decisions that information is supposed to support.
A23: Capacity analysis	A231: Work capacity analysis	The output of work capacity analysis is the order of magnitude and type of work to do to satisfy work specifications, at subsystem level.
	A232: Resource capacity analysis	The output of resource capacity analysis is the order of magnitude and type of resources to satisfy resource specs at subsystem level.
	A233: Decision capacity analysis	The output of decision capacity analysis is the order of magnitude and type of decisions that the active resources must face at subsystem level in the light of the enterprise specs (i.e. strategic or operational), horizon of validity, and revision periods. The required decisions are elicited and validated against the required work, resources, and information elements.
	A234: Information capacity analysis	The output of information capacity analysis is the order of magnitude and types of information needed to capture, store, process, and transfer, at subsystem level.

Table A2: Levels 2 and 3 of the Analysis Activity

Activity	Sub-activity	Description
A24: Structure analysis	A241: Work structure analysis	Work structure analysis has as output the advantages and disadvantages of alternatives for aggregating the element work under specified criteria (e.g. by functional specialization, geography, products, technology).
	A242: Resource structure analysis	Resources structure analysis has as output the advantages and disadvantages of alternatives for aggregating resources under specified criteria (e.g. by functional specialization, geography, products, technology). It considers alternatives for outsourcing or cultivating a resource (a machine, a worker, or a computer system including an ERP system). During resources structure analysis the following occurs: assigning resource classes to potential organizational units (e.g. grouping of resources into cells, shops, departments, plants, divisions); assigning classes of resources to classes of roles; and assigning resource classes to the three subsystems that make up the enterprise (physical, information, and management subsystems).
	A243: Decision structure analysis	The output of decision structure analysis is the set of advantages and disadvantages of alternatives for aggregating decisions under specified criteria (e.g. by functional specialization, geography, products, technology), which lead to scenarios of decision centers, decision roles, span of control, responsibility and authority.
	A244: Information structure analysis	The output of information structure analysis is the set of advantages and disadvantages of alternatives for aggregating information under specified criteria (e.g. by functional specialization, geography, products, technology), and the system model and its evaluation against specifications. This activity uses high level entity-relation diagrams (data), considers the resources and the network (locations and their linkages) related to the IS.

Table A3: Levels 2 and 3 of the Design Activity

Activity	Sub-activity	Description
A31: Strategy design	A311: Work strategy design	The output of work strategy design is the functional design, that is, operational strategies that define how the work will be carried out throughout the enterprise.
	A312: Resource strategy design	Resource strategy design has the operational strategies, from work strategy design, as input, because organization must fit the task (Drucker, 1999). The outputs of resource strategy design are the resource hierarchy (e.g. company, division, plant, department, section, group, and individual), relationships and fundamental and incidental interactions among resources, and layout for the physical system. Resource strategy design identifies the resources (e.g. human, technology, financial) needed to perform the work and decision elements and to produce the selection of products or services specified in the strategy. Alternatives for the major resources are evaluated and selected for later implementation. The level of automation (labor intensity vs. use of manufacturing and information technology) is defined.
	A313: Decision strategy design	Decision strategy design has the operational strategies from work strategy design as input. The output of decision strategy design is the set of decisions that satisfies the enterprise strategic specs and contributes towards the enterprise performance, revision period and horizon of validity of this set of decisions, and the roles that will carry out these decisions.
	A314: Information strategy design	Information strategy design has work, resources, and decision strategy designs as inputs. The output of information strategy design is the IS development methodology, the languages, and the general technology of the future information system. No specific supplier is considered yet. An agreement of terminology and representations (modeling) must be reached, so everyone within the enterprise has the same understanding of the concepts managed. Performance metrics that will provide feedback need to be defined. The other main output is the architectural design of the computer information system.
A32: Competency design	A321: Work competency design	Work competency design has as output the functional capabilities, which include productive, maintenance, administrative, marketing, and control. Work competency design defines the work to do and the workflows within the system and with external customers and stakeholders.

Table A3: Levels 2 and 3 of the Design Activity

Activity	Sub-activity	Description
A32: Competency design	A322: Resource competency design	Resource competency design has as input the functional capabilities and flows, and decision capabilities and flows. The outputs of resource competency design are the core competencies that deliver value to customers, the roles that provide those capabilities (e.g. productive, maintenance, administrative, marketing, control), the competencies to develop over time or to outsource, the evaluation and selection of resources that can provide core competencies, and the resource flows within the system and with external customers and stakeholders. Resource competency design selects the technology to use throughout the system for manufacturing, IT infrastructure and services, including operating system, database management system, application integration services (e.g. CORBA, DCOM), Web services (e.g. navigation, GUI and GUI customization, browsing, loading/downloading services).
	A323: Decision competency design	The output of decision competency design is the set of competencies needed to support work, which will take the form of the decision state space. Decision flows depend on competencies. There are two types of decision flows. One type is made up of the decisions that control the actual movement of resources, information, or work. This set of decisions is called behavioral rule set in CIMOSA (Kosanke et al., 1999) and operating system in GIM (Vernadat, 1996). The other type is the set of decisions passed from higher level to lower levels in the organization structure to direct and coordinate the system (e.g. guidelines, constraints, and time frames useful for management purposes).
	A324: Information competency design	It has as input the work, resource, and decision competency design. It has as output the actual IS design and how the IS handles the flows of data and information. It can use data models, sequence diagrams, activity diagrams, collaboration diagrams, flow diagrams, and state charts. It addresses how the IS will support work and decisions. Procedures for information exchange among enterprise elements and specific internal and external communication channels are defined for the enterprise transactions and management requirements. Interfaces among resources for handling the input and output of data are designed. Define physical means (i.e. hard copies, invoices) and electronic flows for the movement of information among resources. Information flow handles schedules, timing, and rules for the flow of control and the administration of information queues as well. Information flow supports the formalization of business rules. Business rules result from the cardinality and association relations between enterprise elements, from pre and post conditions when there is a dynamic behavior, or from mathematical calculations (Rodriguez et al., 2004).

Table A3: Levels 2 and 3 of the Design Activity

Activity	Sub-activity	Description
A33: Capacity design	A331: Work capacity design	The output of work capacity design is the amount of the work element to be performed by system, or subsystems, in terms of business processes, activities, tasks or any other aggregation of the work element.
	A332: Resource capacity design	The inputs of resource capacity design are the work capacity design and the decision capacity design. The output of resource capacity design is the set of capacities at system and resource levels and the selection of specific technologies and resources (HR, manufacturing, and IT) that perform the work and decision elements. The set of selected resources represent a design solution (Chen et al., 2003). The selection of resources and a specific technology represent a major milestone in enterprise design.
	A333: Decision capacity design	The output of decision capacity design is the specification of the necessary decisions, amount and types, that will direct and coordinate resources for the execution of the work element. Decision capacity design represents the size of the management subsystem, which in turn influences the overhead or indirect costs of the enterprise system.
	A334: Information capacity design	Information capacity design has as output the capacity of the information system for capturing, storing, processing, transferring, displaying, and managing data in order to support work transactions and managerial work.
A34: Structure design	A341: Work structure design	The output of work structure design is the work hierarchy or work break down structure: program, project, deliverable, task, sub-task, operation, and work step; and work classifications (e.g. managerial, technical). The work structure facilitates the assignment of the work elements to the resources responsible for their execution.
	A342: Resource structure design	The output of resource structure design is the resource architecture, indicating the distribution of sets of resources across the enterprise and their relationships. It includes the resources hierarchy, e.g. company, division, plant, department, section, group, individual, resources relationships, given by their roles, authorities, and responsibilities. Resources can perform one or more roles. When the responsibility for all the enterprise elements (work, decision, information, and resources) needed to perform a business process is assigned to one organizational unit, that organizational unit has autonomy (Chen et al., 2003).

Table A3: Levels 2 and 3 of the Design Activity

Activity	Sub-activity	Description
A34: Structure design	A343: Decision structure design	Output: hierarchy of decisions, e.g. strategic, tactic, operational. It ultimately establishes the command and control hierarchy by assigning decisions to roles. It must articulate what role is specifically responsible for formulating decision frameworks and achieve specific objectives (Molina, 2003). A classification with four types of decision was proposed by Olegario and Bernus (2003): • High level decisions, mostly strategic and tactical. The focus here is not in designing how each decision is going to be made (high level decisions tend to be non-procedural), but on specifying that those decisions need to be made, by what roles, and with what interactions with other enterprise elements (flows). Autonomy of resources is specified at this level. • Time-based decisions or management decisions, expected to happen at certain intervals. • Transactional oriented decisions: they control the actual execution of work and deal with real time and day to day decisions. These are triggered by expected events. • Unexpected decisions: these are triggered by unpredictable events. The general rule is that this kind of decision is to be addressed by the lowest decision level with the authority to reconfigure the resources or the work necessary to face the unexpected event. This decision level may also choose not to address the event.
	A344: Information structure design	The output of information structure design is the information static structure, including the classes of data (label, attributes, font, length, data type, visibility, expiration date, data dictionary) in a conceptual scheme of the database (i.e. class, object, component, and deployment diagrams of subsystems or the entire system, data integration), its processes (programming of functions and entire applications that will carry out work), the design of the physical network infrastructure including security. In terms of the Zachman's framework (2003) the information structure design corresponds to the perspective of the technology model of the IS. The information structure supports different functions, e.g. production data organized by work order; engineering data by operation type.

Table A4: Levels 2 and 3 of the Implementation Activity

Activity	Sub-activity	Description
A41: Strategy implementation	A411: Work strategy implementation	The output of work strategy implementation is what to do, by translating the designed functional strategies (e.g. marketing, finance, HR, manufacturing) into specific actions, stating how, where, and when to perform aggregations of the work element. All the work across the enterprise must fit the corporate, competitive, and functional strategies and must be coordinated to achieve the desired objectives.
	A412: Resource strategy implementation	The outputs of resource strategy implementation are the selection of main resources and suppliers for the designed technical solution, the allocation of financial resources, the definition of how the resources will be acquired, the target capital structure (i.e. percentage of the total assets that will be acquired with own resources, percentage funded by debt, or by suppliers, or by other creditors), and budgets and working capital to support the achievement of detailed objectives. The action plans for training human resources and deployment of all resources are devised.
	A413: Decision strategy implementation	The output of decision strategy implementation is the distribution of authority and responsibility for decision making, and how performance measures will provide feedback to each hierarchical level in the organization.
	A414: Information strategy implementation	Output: development methodology, languages (metamodels or glossary for the appropriate conveying of meaning), tools, and the general information and communication technologies. Achieve agreement of terminology, representations (modeling), and feedback metrics. Information strategy is communicated enterprise wide. Tools for defining processes, requirements and design modeling, controlling versions, managing change (track, prioritize, assign, and track progress of software change orders), project management and scheduling, and groupware and repository tools are selected (Nalbone et al., 2004). Information strategy implementation defines the implementation environment, tools, programming languages (e.g. C++, Java), and guidelines for code structure, user interface and usability, documentation, library of standard components. Choosing an IS development methodology implies following some practices, as those of RUP, which is based on the following practices: iterative development; user requirements management; use of reusable components; visual modeling; quality verification at each development phase; and control over change requirements. The way in which performance measures will be gathered and used is defined.
A42: Competency implementation	A421: Work competency implementation	The output of work competency implementation is an action plan for how to get elementary competencies and how to use them to devise work elements, procedures, tasks, activities, business processes or any other aggregation of work, for the purpose of performing business transactions or managerial oriented duties.

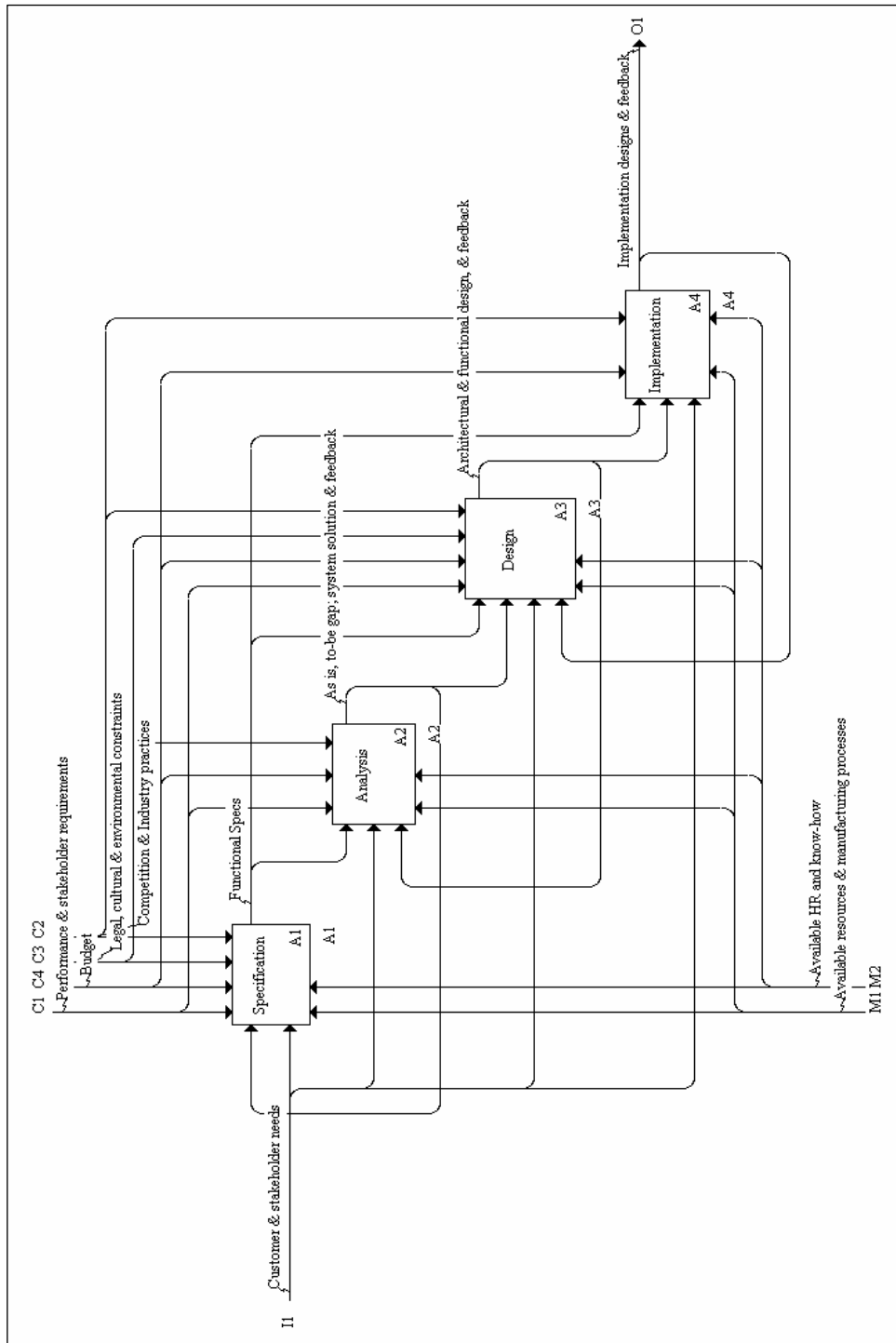Table A4: Levels 2 and 3 of the Implementation Activity

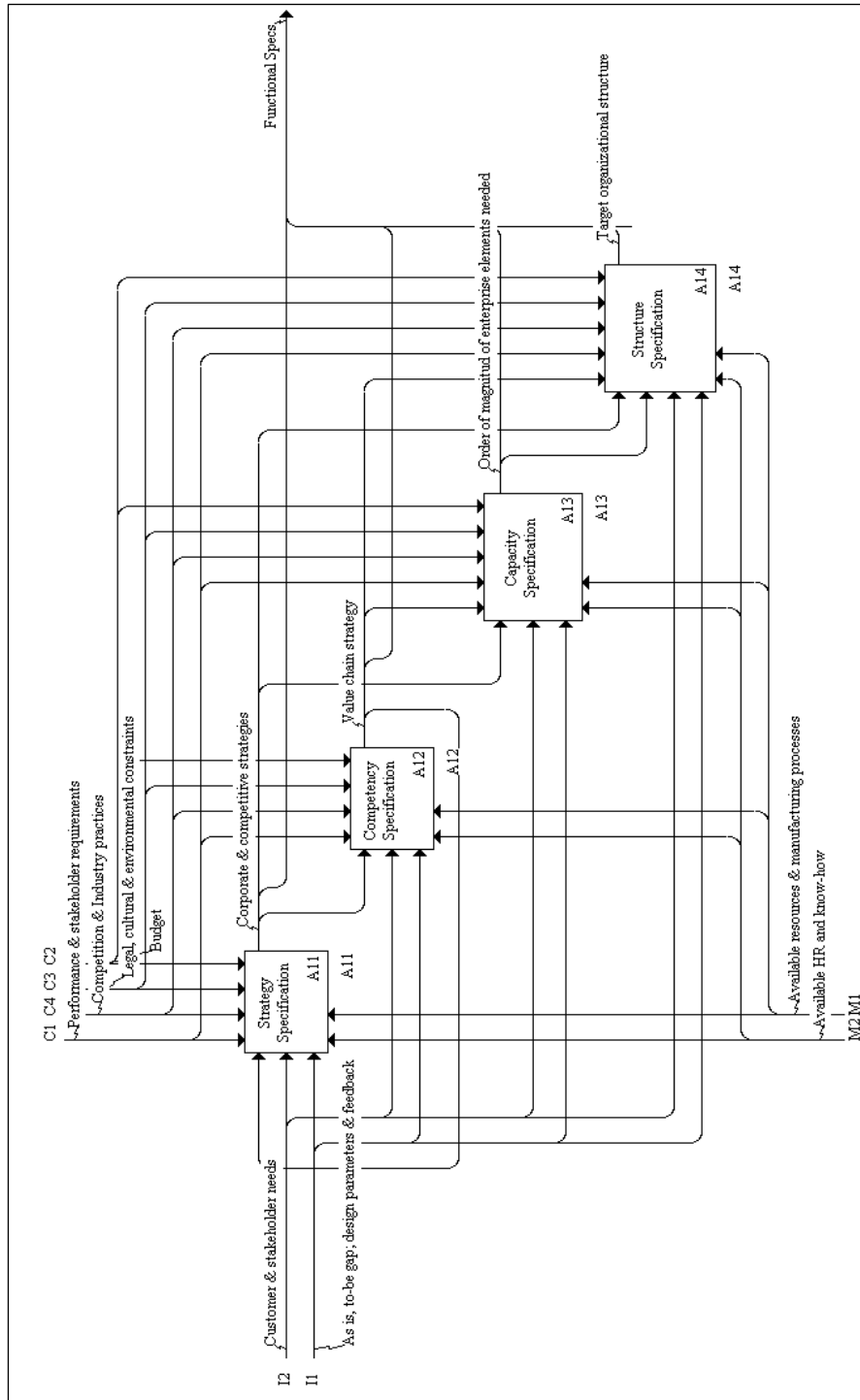| Activity	Sub-activity	Description
A42: Competency implementation	A422: Resource competency implementation	The output of resource competency implementation is an action plan for the selection of actual resources that provide the required capabilities to perform work, a hiring and training plan for in-house resources, and the development of an action plan for competency acquisition from external sources (supply chain, virtual enterprises). A derived output of competencies implementation is the resources flow, the actual routes and movement of resources. If there is a flow between resources there is a coordination need.
	A423: Decision competency implementation	The output of decision competency implementation is a plan for actual deployment of decisions. Both, the implementation of control decision (i.e. sequencing, timing, rules) and the implementation of decision frameworks deal with defining the actual originators and recipients of the decision flows.
	A424: Information competency implementation	The output of information competency implementation is a plan for the actual building of the IS, alpha and beta testing policies and guidelines, expected system and subsystem responses and performance, IS deployment and maintenance, debugging policies and guidelines, and documentation and user training. Competencies define flows. From an organizational perspective, information flows can represent reporting channels and authority channels; reporting channels convey information from lower levels about transactions and events; authority channels convey decisions (Olegario & Bernus, 2003). This is why competencies constrain structure and capacities.
A43: Capacity implementation	A431: Work capacity implementation	The output of work capacity implementation is a plan for how to realize work elements, in-house or form subcontractors.
	A432: Resource capacity implementation	The output of resource capacity implementation is a plan for getting the actual resources that perform work, their deployment/installation (or upgrade) and training needs, and the development of contracts with subcontractors and suppliers. Resources capacity implementation defines specific manufacturing equipment, information technology hardware, software applications to acquired or develop, human resources to hire, vendors, outsourcers, and any other needed resources are chosen. Resources capacity implementation establishes where all the resources are to be put in place (layout) for the business processes to be tested (verified) and validated. It includes how to acquire or develop documentation for operation and maintenance of IT and manufacturing resources.

Table A4: Levels 2 and 3 of the Implementation Activity

Activity	Sub-activity	Description
A43: Capacity implementation	A433: Decision capacity implementation	The output of decision capacity implementation is a plan for getting the actual capacity to perform decisions, a training plan when needed, and the deployment or distribution of authority and responsibility for decision making. The management subsystem is the one that make things happen in the enterprise and it is made up of the decision element. It is the middle layer of the enterprise system, it is not observable by itself but in the physical system, when resources execute decisions, or when some information is stored or transmitted. The management subsystem is also called decisional structure, system of management, system of coordination, or management, command and control (Olegario & Bernus, 2003). The management subsystem supports the planning, coordinating, directing and controlling of the physical subsystem, and it is supported by the information subsystem. Because it does not add direct value it is part of the enterprise overhead or indirect costs. The management subsystem is purposively separated from resources; resources come and go, the management subsystem must remain in place.
	A434: Information capacity implementation	The output of information capacity implementation is a plan for putting in place the enterprise information system, which facilitates the coordination, cooperation and systematic information exchange of the information element among resources. Information capacity implementation is concerned with the actual testing (alpha and beta testing), deployment or switch-over process for the information system. It includes subcontractors when needed. It also deals with documentation for future operation and maintenance, user training, validation (satisfaction of user requirements), assignment of user privileges, and development of the supporting infrastructure (e.g. organization-wide models, standards) that will give support for using and maintaining the resulting information system. The number of resources and their needs for information gives an order of magnitude of the number of interfaces needed and consequently a order of magnitude of information capacity.
	A434: Information capacity implementation	The information system is the inner layer of the enterprise system. The IS supports the work carried out by resources, by storing data, and providing information and the necessary linkages (human-IT or manufacturing-IT interfaces) between interacting resources. These interfaces facilitate the providing, sharing, and managing (create, read, update, delete) of data. In the management system, the IS supports the decisions that need to be made, it gathers and distributes information about transactions and performance feedback. Rodriguez et al. (2004) suggests that each information element must contribute to some business objective. Information capacity implementation is extensive and time consuming. Rodriguez et al. (2004) indicates that the implementation of an information system includes:

Table A4: Levels 2 and 3 of the Implementation Activity

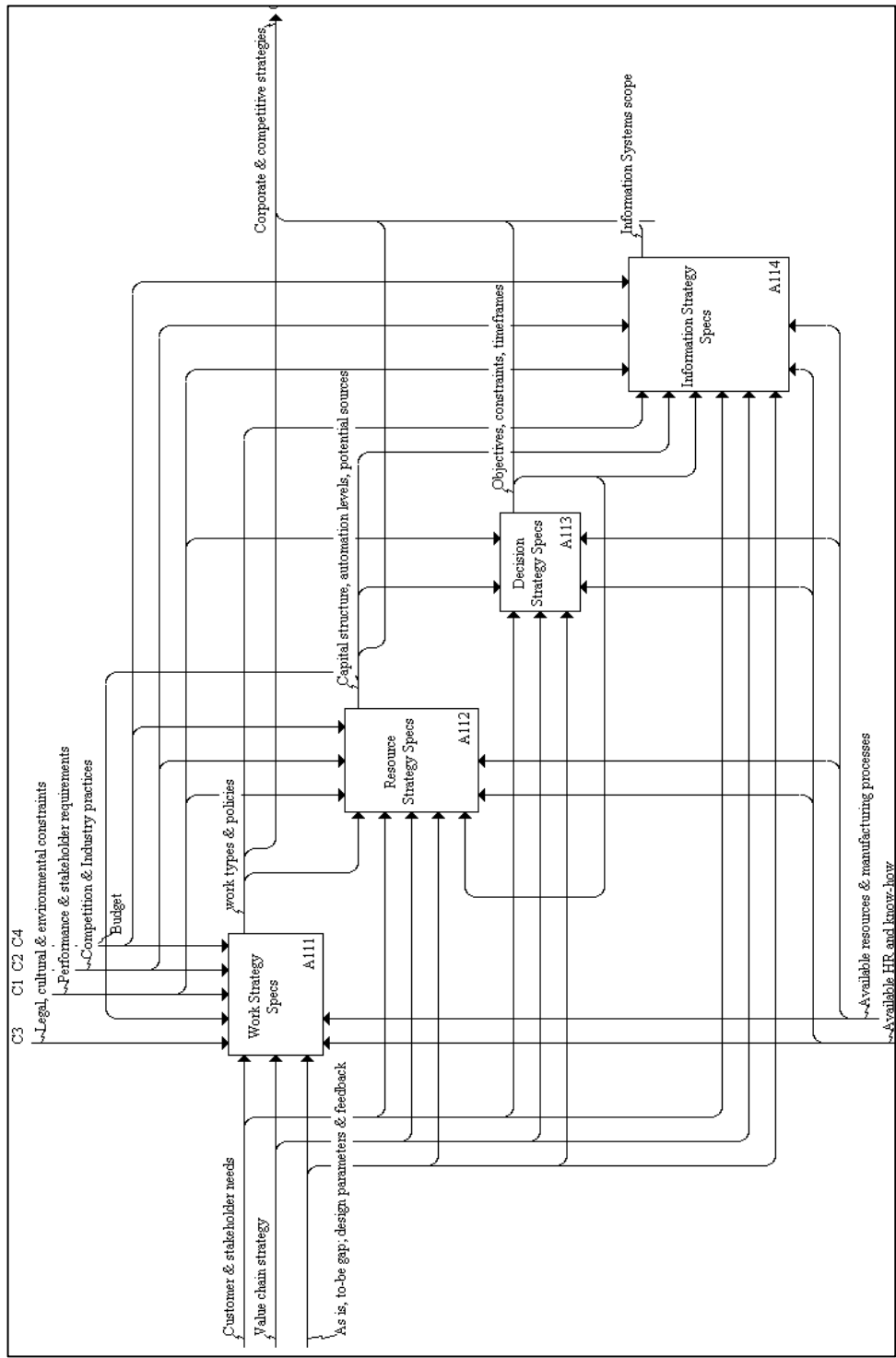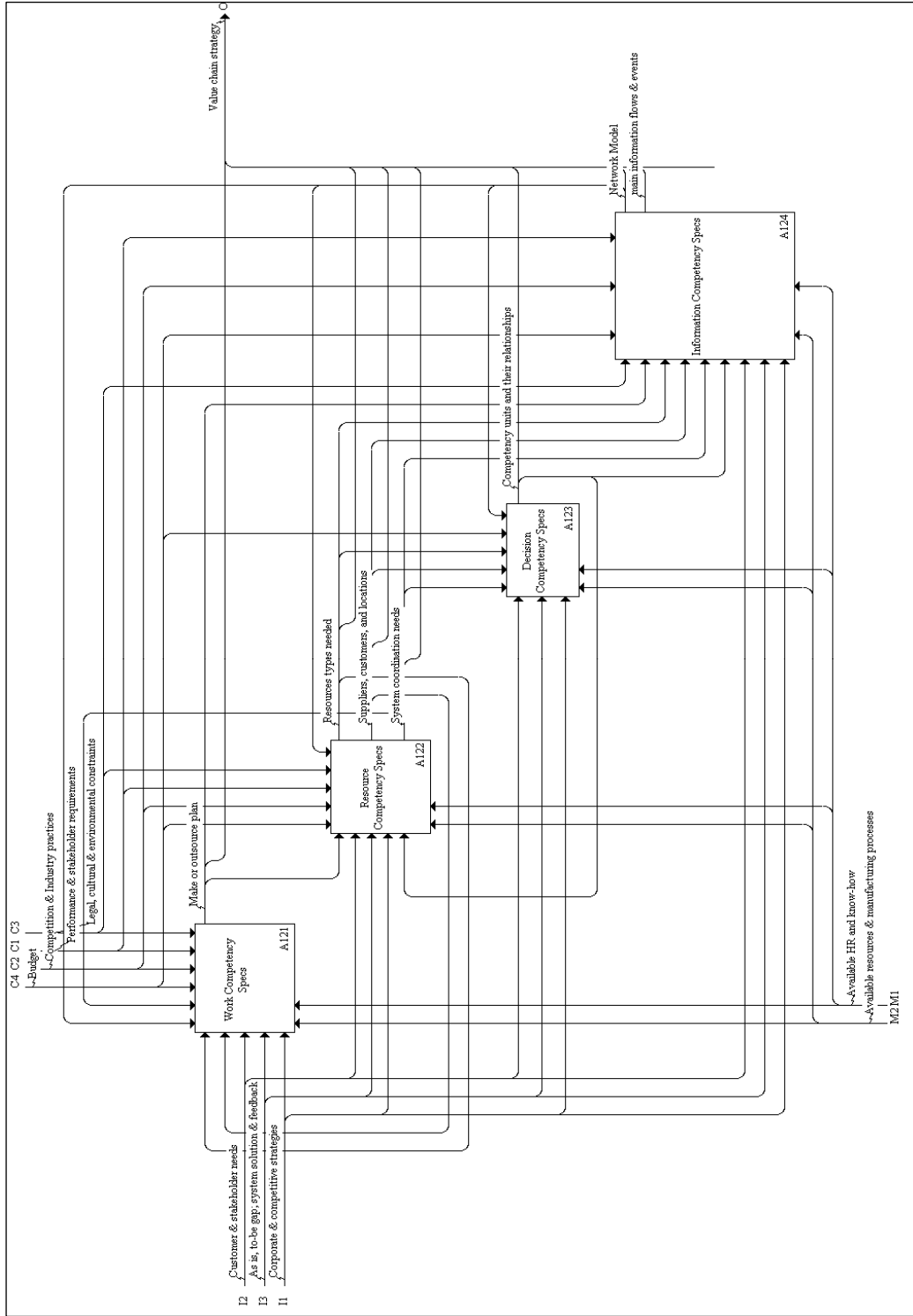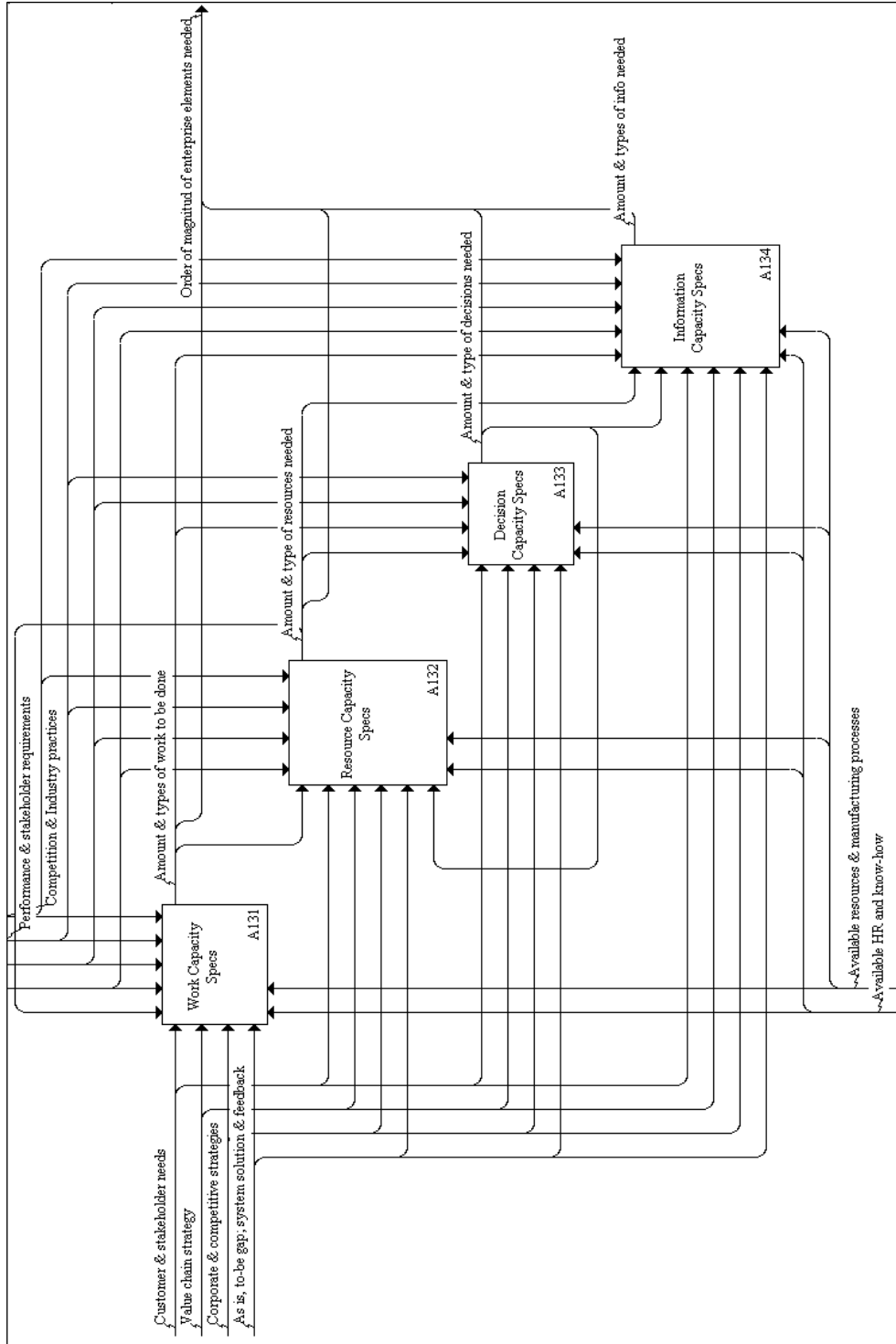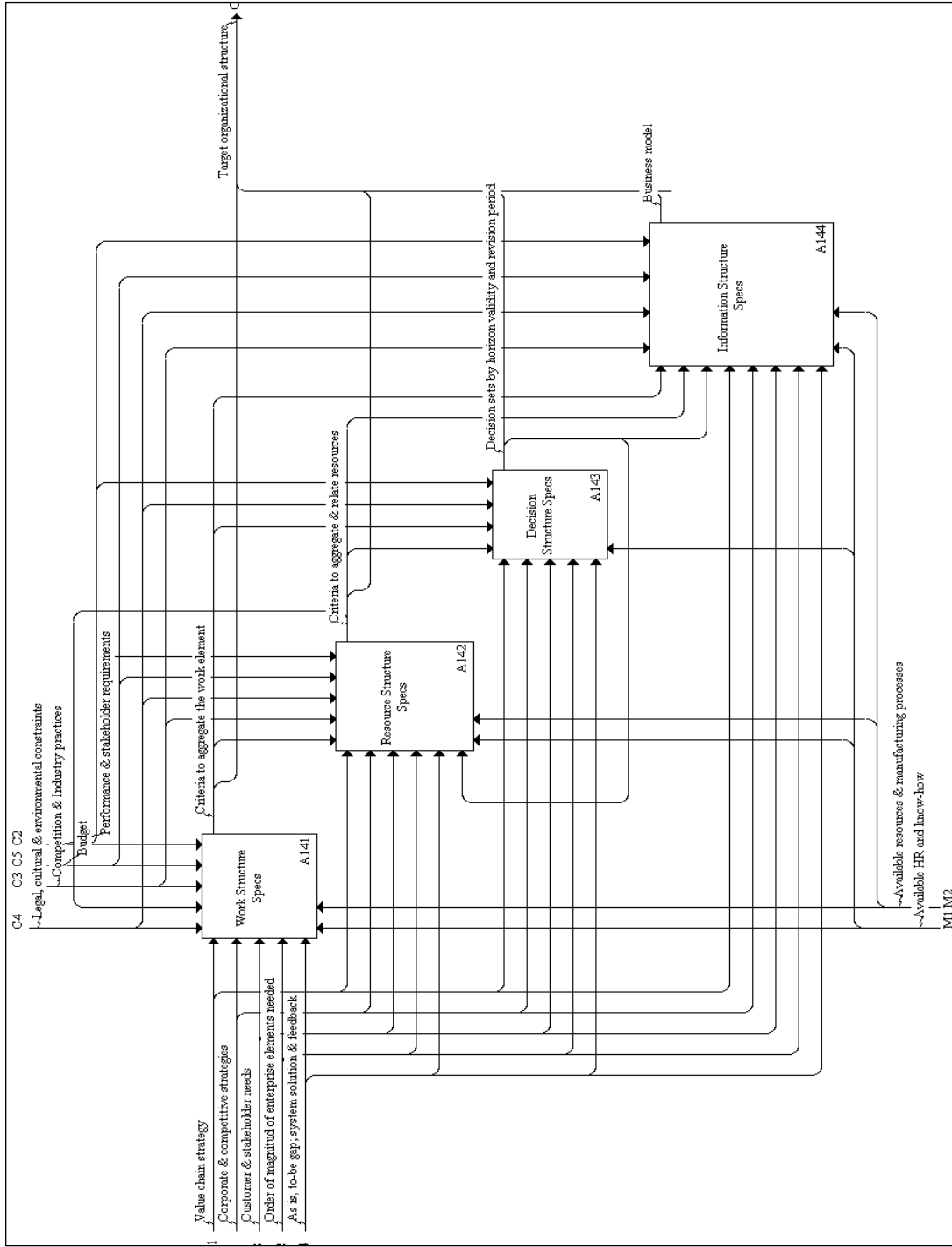| Activity	Sub-activity	Description
A43: Capacity implementation	A434: Information capacity implementation	• Applications: Application security and application integration, transaction support (network and internet infrastructure and services, such as session administration), and communication method. Institutional rules. Workflow (transaction life cycle). Deliverables administration (interfaces, links, event notification). • Infrastructure architecture: physical, security, operating system, DBMS, programming language, development tools (requirements administration, analysis and design modeling, change administration). • Maintenance: infrastructure, database, and applications maintenance; including maintenance manuals and training manuals for operation and maintenance. • Testing: Unit, component, integration, system.
A44: Structure implementation	A441: Work structure implementation	The output of work structure implementation is a plan for the breakdown of work until reaching individual work elements for the purpose of being executed by specific resources, and integrating them in subsystems.
	A442: Resource structure implementation	The output of resource structure implementation is a plan for the actual organization structure for all the resources, until individual resources are assigned work and decision to perform, authority, responsibility, and roles. Resource structure implementation plays the role of a deployment and installation design, grouping resources in units, department or other subsystems. Provision for dynamic allocation of resources to roles may occur.
	A443: Decision structure implementation	The output of decision structure implementation is a plan for the implementation of the managerial system. It defines roles, relationships between roles (e.g. cooperation, subordination, authority, and responsibility), positions, and authority to distribute decisions, objectives, and time frames toward lower levels in the organization structure and what resources are assigned to roles (for execution, coordination and control of the enterprise). For highly dynamic environments, mechanisms for authority allocation for new situations (not included in the original design) are made.
	A444: Information structure implementation	The output of information structure implementation is a plan that establishes how sets of information elements are grouped and deployed, which becomes reports, forms, and databases, and the relationships among them. Sets of information elements are called components. Components are self contained processes or services with predetermined functionality that may be exposed through a technology interface (OMB, 2004). Components need to interoperate, so how to integrate them is part of the output too.

Node A0

Node A1

188

C3   C1 C2 C4

Legal, cultural & environmental constraints

Performance & stakeholder requirements

Competition & Industry practices

Budget

work types & policies

Customer & stakeholder needs

Value chain strategy

As is, to-be gap; design parameters & feedback

Corporate & competitive strategies

Information Systems scope

Objectives, constraints, timeframes

Capital structure, automation levels, potential sources

Work Strategy Specs
A111

Resource Strategy Specs
A112

Decision Strategy Specs
A113

Information Strategy Specs
A114

Available resources & manufacturing processes

Available HR and know-how

Node A11

189

Node A12

Work Capacity Specs — A131

Resource Capacity Specs — A132

Decision Capacity Specs — A133

Information Capacity Specs — A134

Customer & stakeholder needs

Value chain strategy

Corporate & competitive strategies

As is, to-be gap; system solution & feedback

Performance & stakeholder requirements

Competition & Industry practices

Amount & types of work to be done

Amount & type of resources needed

Amount & type of decisions needed

Order of magnitud of enterprise elements needed

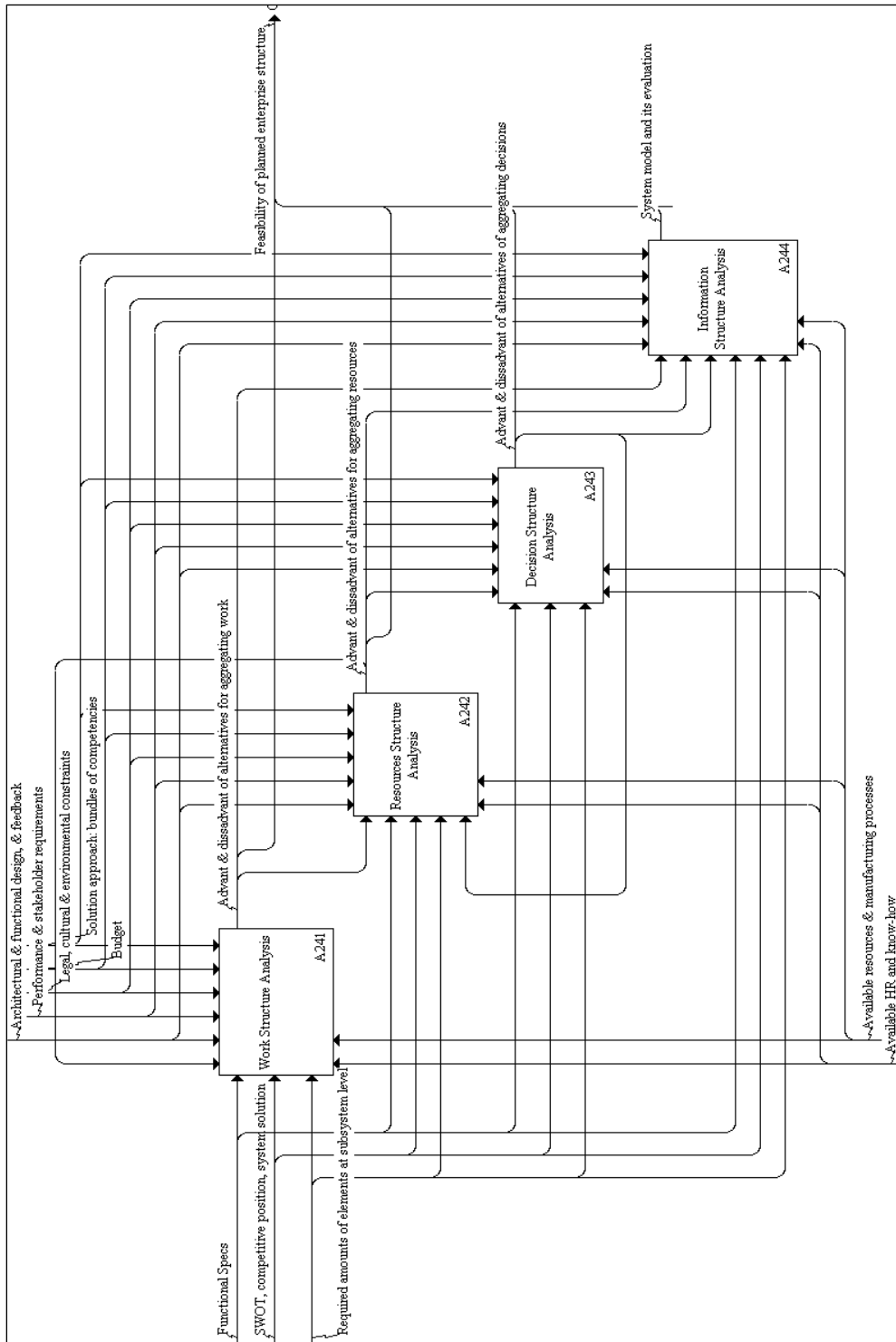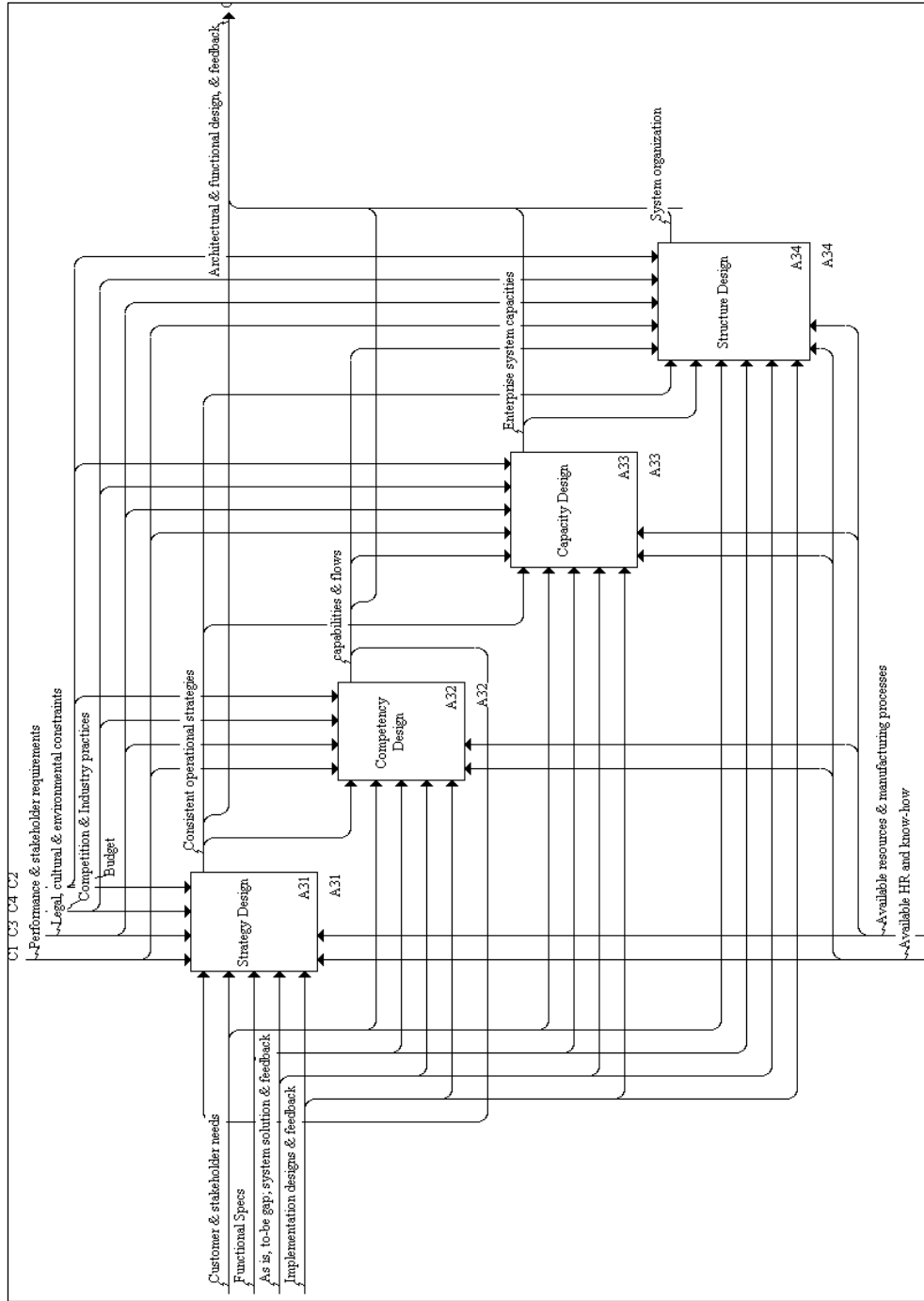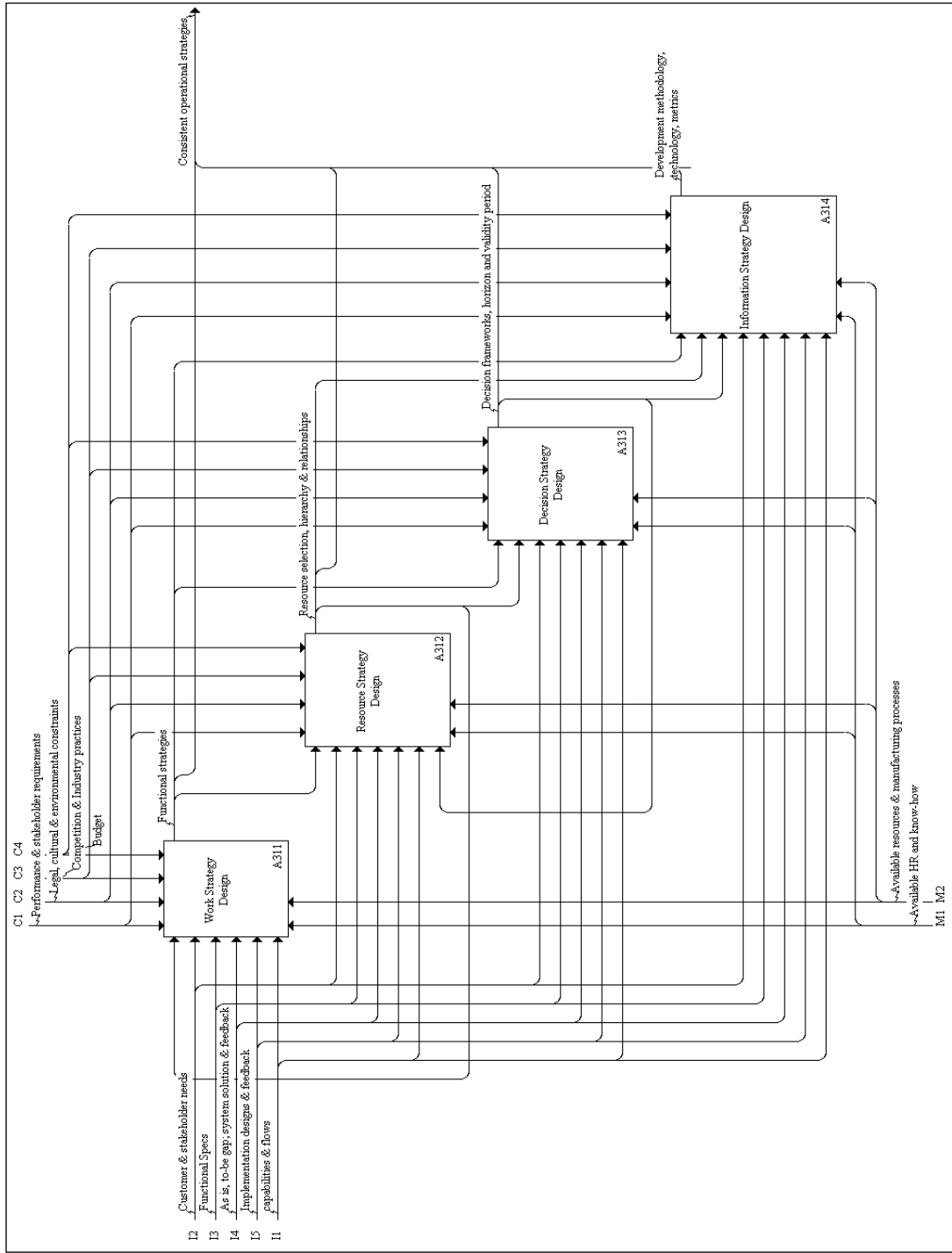Amount & types of info needed

Available resources & manufacturing processes

Available HR and know-how

Node A13

191

Node A14

Node A2

Node A21

194

Node A22

195

C5 C2 C3 C4 C1

Architectural & functional design, & feedback

Performance & stakeholder requirements

Legal, cultural & environmental constraints

Solution approach: bundles of competencies

Budget

Required amounts of elements at subsystem level

Required types & amount of decision, horizon of validity and revision periods

required amount & type of info

Resources to support work, decision & information

Required amount of work

Customer & stakeholder needs

Functional Specs

SWOT, competitive position, system solution

Available resources & manufacturing processes

Available HR and know-how

Work Capacity Analysis — A231

Resource Capacity Analysis — A232

Decision Capacity Analysis — A233

Information Capacity Analysis — A234

Node A23

196

Node A24

197

Node A3

Node A31

C1 C2 C3 C4

Performance & stakeholder requirements

Legal, cultural & environmental constraints

Competition & Industry practices

Budget

Consistent operational strategies

Development methodology, technology, metrics

Decision frameworks, horizon and validity period

Resource selection, hierarchy & relationships

Functional strategies

Work Strategy Design
A311

Resource Strategy Design
A312

Decision Strategy Design
A313

Information Strategy Design
A314

Customer & stakeholder needs

Functional Specs

As is, to-be gap; system solution & feedback

Implementation designs & feedback

capabilities & flows

I2
I3
I4
I5
I1

Available resources & manufacturing processes

Available HR and know-how

M1 M2

199

Node A32

Node A33

201

Node A34

202

Node A4

203

Node A41

Node A42

205

Node A43

206

Node A44

207

VITA

OSCAR ALEJANDRO SAENZ

EDUCATION

04-1986	BSc. Industrial Engineering Universidad Centroamericana, Managua, Nicaragua
06-1987	Master of Business Administration (MBA) INCAE Business School, Alajuela, Costa Rica.

PROFESSIONAL EXPERIENCE

1989-2000	Associate Project Manager EuroConsult, S.A., Managua, Nicaragua
1997-1999	Visiting Professor MBA, Universidad Centroamericana, Managua, Nicaragua
2001-2004	Adjunct Instructor, Teaching / Research Assistant Industrial & Systems Engineering Department Florida International University, USA

PUBLICATIONS AND PRESENTATIONS

Giachetti, R., Martinez, L., Saenz, O., and Chen, C-S. "Analysis of the Structural Measures of Flexibility and Agility Using a Measurement Theoretical Framework." International Journal of Production Economics 86 (1): 47-62, 2003.

Saenz, Oscar, and Chen, Chin-Sheng. "Identification of Modeling Areas for Enterprise Systems Engineering." Institute for Operations Research and Management Sciences (INFORMS) Annual Conference, Denver, 2004.

Saenz, Oscar, and Chen, Chin-Sheng. "Towards a Framework for Enterprise Systems Engineering." 2nd Latin American and Caribbean Consortium of Engineering Institutions (LACCEI) Conference, Miami, FL, June 22, 2004.