CHAPTER ONE
INTRODUCTION

1.1 Background of Study

Enterprises globally are undergoing business transformation. In order to take advantage of this transformation and gain competitive advantage, enterprises all over the globe have been reinventing their business models. These business trends have resulted in adoption of innovative technologies and increased use of modern technology to communicate and collaborate. Today's market conditions demand better alignment and synchronization of business strategy, processes, and technology. Unfortunately, business and Information Technology (IT) architectures that are supposed to support innovation and competitive advantage are often disconnected, thus severely constraining an organization's agility. The successful organization of the future will be an engine of continuous transformation that adjusts to offer solutions to its constituents at a lower cost and faster time-to-market than its competitors (Schekkerman, 2003). This transformation requires an integrated Business and Technology Architecture – Integrated framework that not only dynamically aligns business and technology components and processes within the organization and across its value chain, but also supports the organization for long term, cost-effective growth, manage IT system complexity and scale of change in the business. This transformation will be supported by a single enterprise architecture model called An Integrated Model for Small and Medium Scale Enterprise Systems (AIMES) in which business architecture is related to information and Information Technology architecture.

In the early days of computing, technology simply automated manual processes with greater efficiency. As technology evolved, new innovations enabled new capabilities and processes in the enterprise driven by IT. Gradually, IT changed the business but not necessarily in alignment with the business strategy. This lack of alignment resulted in significant waste of resources and missed opportunities, and placed the organization in a competitive disadvantage in the market. Over the years, IT has evolved from delivering point solutions to a complex, interrelated landscape of applications, interfaces and infrastructure that support the business processes of an organization and the productivity of its people. More recently, this has started to include an architectural view of business change, so that Business and IT function seamlessly to deliver the goals of the business. Architecture has always played a role in the development of systems. However, until the early to mid 90’s (Capgemini, 2007), it was almost exclusively used in technical infrastructure, and commonly referred to as Systems Architecture. As applications and systems increased in number and complexity, the need for a clear and consistent view of the complete picture, together with a structured approach to integration, became apparent. Gradually, the term architecture was extended to include all areas involved; initially ranging from technical infrastructure to information systems, and then towards information, processes and business. More recently, the differences between architecture at an Enterprise level and at a Solution (or project) level have become more clearly recognized and defined: Architecture at the Enterprise level is oriented to the overall business, information and systems landscape, whereas at the project or Solutions level, architecture is more focused on a definition of solution direction and high level design (Capgemini, 2007).

Uncontrolled growth of information systems and technology in the late 1990s (often as a result of decentralized decision making) resulted in information and systems landscapes becoming complex, costly and difficult to manage. As a result, responding quickly and efficiently to new business challenges has become increasingly difficult. Changes in the business affect various aspects of operations: operational scenarios, business processes, policies and important business metrics. All these changes, in turn, have an impact on the systems that are used to automate business operations. Often changes in the company’s technology infrastructure that underpins its business systems are needed to achieve the desired shifts in business model and operations. Managing change of enterprise scale and complexity requires a structured approach that can holistically cover all impacted areas of the business and plan for major changes in business capabilities to achieve strategically relevant outcomes. Architecture is critical to managing this complexity (Pessi, 2010).

The word “architecture” is among the words, which nowadays has been using much in the world of information technology. According to Pessi and Magoulas (1998), the word “architecture” is going to replace the word “structure” in overall terms. Hugoson et al.

(2008) mentioned that since 1970's, organizations are spending huge amount of money for building new information systems. Yet there have been some obstacles in that respect which are:

i) The fast growing amount of systems which in most cases are integrated in ad hoc manner have been expected to increase the cost and complexity of information systems.
ii) Organizations were finding it more and more difficult to keep these information systems in alignment with business needs.
iii) The role of information systems has changed during this time, from automation of routine administrative tasks to a strategic and competitive weapon and the nature of applications has evolved dramatically, from the simple batch systems of the 1960s to today's networked distributed apps, which are capable of handling much higher transaction volumes.
iv) Enterprises have not only demanded more applications but, increasingly, faster time to market and responsiveness as well as greater agility and flexibility — often to support dynamic business evolution where requirements are not as well defined as previously. In other words, business has demanded that IT get off the critical path of business change and evolution.
v) To support and enable new kinds of applications on far more complex hardware configurations, middleware has grown from almost nothing to today's high-performance, high-capability middleware products.

In providing solution to these obstacles, a new field of research was born that soon become known as Enterprise Architecture. Enterprise architecture is defined as the process of translating business vision and strategy into effective enterprise change by creating, communicating, and improving the key requirements, principles, and models that describe the enterprise's future state and enable its evolution (INCOSE, 2005). During these developments in the field of Enterprise Architecture, some frameworks were created like Zachman, The Open Group Architecture Framework (TOGAF), and Federal Enterprise Architecture Framework (FEAF). The expressed reason for having Enterprise Architecture is that it provides blueprints for organization and delivers an approach with a set of design principles, methods and models that can be used to design and realize the structure of an organization on enterprise level. Enterprise architecture supplies to people at all organizational level an explicit, common, and meaningful structural frame of reference. Furthermore, it allows an understanding of important facts such as:

(i) What the Enterprise does;
(ii) When, where, how and why it does that;
(iii)What it uses to do it.

In general view and regarding modern organizations, Magoulas et al. (2011) mentioned that having enterprise architecture as a blue print is not just limited to improve competiveness, but also reduces complexity, increases changeability and provide a basis for evaluation.

EA considers an enterprise as a system in which competencies, capabilities, knowledge, and assets are purposefully combined to achieve stakeholder goals. The tangible outcome of this line of reasoning is a blueprint or holistic overview of the enterprise in the form of an integrated collection of models. Hence, architecture can help maintain the essence of the business, while still allowing for optimal flexibility and adaptability (Jonkers et al., 2006).

EA approaches are often experienced as complex, over-engineered, and difficult to implement. Because of the technical detail required for full-scale implementation, EA models tend to become very large, making them more difficult to understand and less effective to reflect on or design enterprises and their supporting systems (Balabko & Wegmann, 2006). Due to their resource poverty, SMEs experience even more difficulties than larger enterprises in employing EA experts or hiring external consultants (Kroon et al., 2012). Yet, as some studies have confirmed, they may encounter several problems if they fail to implement EA (Bidan et al., 2012).

Bernaert et al. (2013) proposed the concept of EA as a good solution to be used for SMEs to solve problems related to a lack of structure and overview. However, EA is still unknown and hardly used in SMEs. A recent exploratory field study by Bernaert et al. (2013) examined 27 SMEs and observed that nearly all of them were missing a clear overview of their business organization and none of them actually were using enterprise architecture. The study concluded that there is a pressing need to develop an EA approach specifically adapted to the SME context, consisting of a metamodel, a method, and software tool support.

The goal of this research is to design and develop an integrated EA model for the implementation of small and medium scale enterprise system. The value of the current research lies in the fact that, to our knowledge, AIMES is the first effort to actually develop an Integrated EA approach specifically adapted to the SME context. The development of the metamodel was guided by the requirements for EA in an SME context.

1.2 Statement of the Problems

There are three major problems that this research will attempt to resolve in the SME;

i. Complexity of Enterprise Level IT Systems: The challenge to build highly complex IT systems, ensure that those systems meet needs of increasingly business processes and do all this in a manner that allows everything to adapt quickly to changing technologies. Most existing enterprise architecture methodologies evolved in a much simpler era. Thus, while they address some of the traditional problems of system integration, they do not address the much more difficult problem of managing today's complexity that arises from changes in business and technology. Organizations were spending more and more money building IT systems.

ii. Poor Alignment of Information Technology to Business Strategy: The challenge to ensure strong alignment between business processes, information system and technologies that support them in the SME. This involves the integration of all aspects of the business and its processes and even across complex multi-partner enterprises using information technology. SMEs were finding it more and more difficult to adapt those increasingly expensive IT systems aligned with business need. Strategic alignment is critical for leveraging organizational performance.

iii. Low Business Value Realization from IT by SMEs: The challenge to build information system architecture that can be used to realize business objectives. Architecture only delivers maximum value when it is an integral part of the overall business change lifecycle. In this way, the whole enterprise (business and technology) can be designed together, informing and supporting the business and IT strategies as well as shaping the business and IT itself. These problems signify that the cost and complexity of IT systems have exponentially increased, while the chances of deriving real value from those systems have dramatically decreased.

1.3. Aim and Objectives of the Study

The main aim of this study is the design and development of an integrated enterprise architecture model for small and medium enterprises (SMEs) to optimize across the SME the fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy.

In order to achieve this aim, the set of specific objectives were to;

i. Create a metamodel for the SME.
ii. Provide an enhanced methodology for enterprise architecture development (method) in SMEs based on continuous improvement process.
iii. Apply the Integrated Model to an SME (Donavan Tiles) to standardize the business processes and streamline operations in order to reduce overhead cost.
iv. Construct a business application classification model for SMEs from the integrated model based on core SME business processes.
v. Implement a prototype integrated enterprise information system for the case SME based on the integrated model.

1.4. Significance of the Study

The integrated model for small and medium scale enterprise system forms the structure for integrating the various architectures of the enterprise namely the enterprise, information, information system, infrastructure and governance and this will be used to;

i. Improve planning - help make more informed IT decisions.

ii. Reduce Complexity – provide a life cycle management by establishing a process that is focused on building, maintaining, acquiring and retiring technology.

iii. Improve IT to Business Alignment – facilitate the adaptation of technology to changing business needs and pressure in enterprise administration.

iv. Standardize business processes and streamline the operations of the SME.

v. Provide a structured overview of the SME.

1.5. Scope of the Study

The subject of integration and enterprise architecture is very extensive and includes a great number of types of systems. The focus of this thesis is on designing archetype enterprise information architecture for the implementation of enterprise systems as an instantiation of an Enterprise Architecture (EA) in terms of a set of software modules, computer platforms, network components, and databases assembled in such a way as to be able to process business transactions and thus meet all the system requirements specified in Enterprise Architecture. Despite the benefits of this research work, it is limited to architecture for integration of enterprise systems. The areas of Intelligent Enterprise Integration and Enterprise Migration and Virtual Enterprise Integration will not be covered in details

1.6. Limitations of the Study

Enterprise architecture is generally unknown and unused in Small and Medium Enterprises (SMEs). In literature, articles about EA for SMEs are very scarce. This is the most limiting factor to the research. Other factors include;

- Limited time in gathering related data/information needed for the design and implementation of the system.
- Finance poses some constraint in the purchase of the software needed for the modeling and designing of the system.
- Human or technical assistance also posed some constraints in extracting information from them.

1.7. Definition Of Terms

Archetype

The original pattern or model from which all things of the same kind are copied or on which they are based; a model or first form; a prototype.

CORBA

Common Object Request Broker Architecture (CORBA). This is a standard developed by (OMG) object management group to provide interoperability among distributed objects. It is the world's leading middleware solution enabling the exchange of information, independent hardware platform, programming language and operating system. It is a design specification for an object request broker (ORB) which provides the mechanism for distributed objects to communicate with one another whether locally or remote devices written in different languages or at different locations on network.

Data Integration

Data integration involves combining data residing in different sources and providing users with a unified view of these data.

Data Sharing

Making use of the same set of data by different system. Ability to allow multiple users to access information at the same time (concurrently).

EC

Electronic Commerce System (e-Commerce). This means information system that processes data and provides information to support the operations and management of an organisations electronic commerce activities. This includes advertising, selling, order entry, order fulfilment, billing, customer support and related activities.

ERP

Enterprise resource planning systems are a one big vendor software package that helps provide best-practice business process functionality running on a single database.

Enterprise

Enterprise is a large business firm, organization or venture.

Enterprise Architecture

A coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organisational structure, business processes, information systems, and infrastructure.

Enterprise Systems

Enterprise systems (ES) are large-scale, integrated application-software packages that use the computational, data storage, and data transmission power of modern information technology (IT) to support processes, information flows, reporting, and data analytics within and between complex organizations.

Enterprise Information System

An Enterprise Information System is any kind of computing system that is of 'Enterprise Class'. This means typically offering high quality of service, dealing with large volume of data and capable of supporting large organisation, 'an Enterprise'.

Integration

Combine (parts) into a whole or it can be to complete by the addition of parts.

Integrated Model

A model that presents some key aspects or component subsystems that is to be brought together into one system. This means also the linking together of the major aspects of different computing systems and software applications physically or functionally**.**

Model

Simplified description of a system to assist calculations and prediction and decision making. It can also be a schematic description of a system, theory or phenomenon that accounts for known or inferred properties and may be used for further studies of its characteristics.

Metamodel

A metamodel is a model that explains the syntax and semantics of Enterprise Architecture models.

Packaged-to-e-business integration

The package application as enterprise resource planning systems are integrated with e-commerce application, for example processes that deal with e-sales, e-procurement and supply chain management are integrated with packaged systems. The organization can use enterprise application integration (EAI) and e-commerce technology for integrating package application and e-business.

SME

(i). A Small and Medium Scale Enterprise (SME) is an enterprise that has asset base (excluding land) of between N5million –N500 million and labor force of between 11 and 300 (CBN).

(ii). An SME can be defined as an organization which has no more than 250 employees and has either an annual turnover of less than Euro 40 million, or an annual balance sheet total less than Euro 27 million.

Supply Chain Management

Supply Chain Management is the management of a network of interconnected businesses involved in the ultimate provision of products and services packages required by end customers. It spans all movement and storage of raw materials, work-in-process inventory and finished goods from point of origin to point of consumption.

System

A system denotes a collection of elements which have mutual relations and which act together to jointly solve some given task (its function). This task could not be solved by any element on its own. {A system could be made out of materials (material system) or from notions, statements, theorems, etc. (ideal system). The elements of a system can itself be systems (subsystems).

System Architecture

A system architecture or systems architecture is the conceptual design that defines the structure and/or behavior of a system.

System Integration

System integration is the bringing together of the component subsystems into one system and ensuring that the subsystems function together as a system or systems integration is the process of linking together different computing systems and software applications physically or functionally.

SOAP

Simple Object Access Protocol. This an xml-based messaging protocol. It defines asset of rules for structuring messages that can be used for simple one-way messaging but is particularly used for performing remote procedure call (RPC) a request/response dialogue.

XML

Extended Markup Language. A meta-language that allows users to define their own customized markup languages especially in order to display documents on the world wide web. XML was designed to transport and store data while HTML was designed to display data. It focuses on what data is while HTML focuses on how data looks.

W3C

World Wide Web Consortium. An international organization for the world wide web. They developed the standard for XML and other areas. It is an organization that houses the standards of web sites and the way they look.

WS

Web Services are typically application programming interfaces (API) pr web API's that can be accessed over a network such as the internet and executed on a remote network hosting the requested services. It is also a group of loosely related web based resources and components that may be used by other web applications over HTTP.

UDDI

The Universal Description Discovery and Integration (UDDI) is a platform independent, extensible markup language (XML) based registry for businesses worldwide to list themselves. It is a platform independent framework for describing services, discovering businesses and integrating business services using the internet.

1.8. Organisation of the Work

The rest of the thesis is organized as follows…

Chapter 2

We looked at the information system architecture as a whole; concepts of enterprise information system, existing architectures, integration approaches and mechanism, and its evolution to extended enterprise system. We also looked at previous works done on enterprise architecture frameworks.

Chapter 3

This chapter explores the research methodology, weaknesses of the existing architecture models, problems of SMEs and the proposed integrated model as a solution to some of the limitations of the existing models.

Chapter 4 Systems Design

Here, we examined the design work. This includes;

i) Develop the AIMES architecture from the views

ii) Develop an integrated Enterprise using a process map.

iii) Build an Integrated system – Total Information Processing System based on the Industry process map.

iv) Test the usefulness of the architecture on a small and medium enterprise.

Chapter five

This chapter shows the implementation of the system using known technologies and Testing of the developed software. The data dictionary, input-output specification, hardware and software requirements

Chapter Six

This chapter tests the architecture and information system using some criteria and derived a conclusion based on some evaluation of the architecture framework.

Chapter Seven

This chapter starts with recommendation, summary, conclusion and future development with references.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

This aim of this chapter is to identify state-of-art in research related to enterprise architectures, enterprise system classification models, enterprise architecture frameworks, SME's and other topics related to this thesis. The models presented in this chapter provide the theoretical background that lead to the proposal and realization of the integrated model for enterprise system.

2.2 Review of Information Systems Architectures

The topic of system architectures has received attention in the Information System literatures for more than two decades. A review of this literature indicates a paradigm shift concerning the type and sophistication of information systems architectures discussed in the Information Systems literature (Ross, 2003). A review of the Information Systems literature, spanning a period of 20 years, revealed a paradigm shift from functional- and business unit-level architectures to enterprise-level architectures. The review indicated an increase in complexity, intensity, and detail with regard to information systems architectures. This study is primarily interested in the enterprise architectures and enterprise architecture frameworks; however, a brief discussion of its predecessors --IS architecture and information architecture-- is conducted to provide a better understanding of the value of the ITA and to develop the ITA maturity into enterprise architecture frameworks.

2.2.1. Framework for Information System Architecture (ISA): A System Development-focused Architecture

In 1987, when the subject of ISA was beginning to receive a great deal of attention, Zachman (1987) set out to define a framework that would facilitate better systems development. Given the "increased scope of design and levels of complexity" of IS implementations, the timing of this endeavor was critical (Zachman, 1987). The primary purpose of the development of this framework was to rationalize the various architectural concepts and specifications to allow for improving and integrating systems development approaches and to establish credibility and confidence in the investment of system resources (Zachman, 1987). Furthermore, he stressed that the difficulty in answering the question "What is information systems architecture?" was partly due to the misconception that there exists a single architecture. He suggested that there existed a set of architectural representations. He further suggested that this set of architectural representations were additive and complementary. Zachman (1987) assertions were supported by Goodhue et al. (1988), whose study implied the existence of multiple architectural foundations.

Drawing from 31 data management case studies in 20 firms, Goodhue et al. (1988) concluded that there was no single dominant approach to improving the management of data. Rather, they found that firms used multiple approaches that differed with regard to business objective, scope, planning methodology, and product. Goodhue et al. (1988) made it clear that the architectural foundations primarily serve as guides for future systems development. Although the architectural foundations lead to well-developed systems, there was still the issue of a lack of data standardization and integration that needed to be addressed. Failure of the architectural foundations to address such concerns could lead to the development of fragmented systems that are not conducive to supporting the business objectives of the firm. Therefore, Hackathorn & Karimi (1988) attempted to address these issues by constructing the overall ISA for the organization. As part of this construction, they compared 26 widely-cited methods for information engineering and concluded that the evolutions to more effective methods of information engineering were needed to align future IS requirements to firms' strategic goals and objectives. Additionally, they saw this evolution as necessary in order to exploit the current IS environment for competitive advantage (Hackathorn & Karimi, 1988).

2.2.2. Information Architecture (IA): An Information Management-focused Architecture

The IA accounted for part of the evolution towards more effective measures of information engineering. The IA is a high-level map of the information requirements of an organization aimed at identifying major information categories in use within an organization and their relationships to the business processes and functions that support the organization (Brancheau et al., 1996). A well developed IA is vital to the successful development of integrated information systems. The purpose of the IA was to facilitate successful implementation of the IS plan (Richardson & Jackson, 1990). Periasamy and Feeney (1997), found that, in most cases, successful implementation of the IS plan led to fulfillment of the IS strategy, which is primarily concerned with aligning IS development with business needs and with seeking strategic advantage from IS applications (Raghu-Nathan et al., 2001).

Sowa and Zachman (1992), and Periasamy and Feeny (1997) all argued that the IA consists of two components -- the data architecture and the application architecture. The data architecture is, in essence, the organization's corporate data model (CDM). The CDM graphically depicts the major entities within the organization and the relationships that exist among these entities. Periasamy and Feeny (1997) stated that this component of the IA is essential for long-term and cost-effective data management. The application architecture graphically depicts the applications that make up an organization's integrated information systems and the data that flows between the applications (Allen & Boynton, 1991; Evernden & Evernden, 2003; Farnum, 2002; Niederman et al., 1991; Periasamy & Feeny, 1997; Pervan, 1998). Periasamy and Feeny (1997) stated that the data architecture component of the IA serves management communication needs during IS planning and later, enables the development of applications in an integrated manner.

Based on their experience working with various organizations over a three year period, Allen & Boynton (1991) presented two approaches towards the development of an information system architecture in which flexibility and efficiency is made possible through systems and data integration. They referred to the two approaches as the "low road" and the "high road." With the low road approach, IS and the management of IS are dispersed throughout the firm. They argued that although IS becomes the responsibility of every operating manager, the approach is more than just one-step beyond decentralization of the corporate IS organization. Additionally, the low road approach views standards as impediments to progress; thus, the role of standards is limited to ensuring the integrity of the internal data exchange processes. With the high road approach, the role of the senior IS executive is expanded as core IS activities are centralized. Applications are designed to be organizationally independent with the expectation of enabling applications to continue meeting the needs of the firm even if the organizational structure changes. The expectations of the high road approach were that it would provide IS efficiency and that are centrally managed, well-integrated IS would enable quick response to strategic challenges. However, the findings were contrary to the expectations and beliefs. Allen & Boynton (1991) found that IS had to be tailored and modified to respond efficiently to local and changing demands. They concluded that even if the high road approach is fulfilled it might never meet the changing expectations of the firm. Although, the two pronged approach towards the management of IS resources appears to have some viability, there is still the issue of implementing an information system architecture that will enable firms to be competitive at any given level.

2.2.3. Information Technology Architecture (ITA): An Enterprise Resource Management Architecture

Whereas the predecessors of the ITA tend to focus more on systems development, data sharing, and systems integration efforts within the organization, this particular architecture tends to focus more on the governance of IT resources within the organization. If implemented as specified, an ITA specifies (a) how and why the pieces of the IT infrastructure fit together as they do; (b) where the pieces of the IT infrastructure go and at what time they are needed; and (c) why and how changes in the IT infrastructure will be implemented. In the context of this study, the detailed definition of ITA is as follows: the plan (or set of plans) that serves as the organizing logic for decisions that pertain to data, applications, IT infrastructure (technical and human), and management responsibilities and strategies (IT and business), captured in a set of policies, procedures, and technical choices that guide and direct the arrangement, development, and accessibility of those elements with the intent to achieve desired business and technical standardization and integration to enable the accomplishment of a firm's business objectives (Allen & Boynton, 1991; Gibson, 1994; Ross, 2003).

As indicated by the comprehensive definition of ITA, an ITA encompasses and extends the ISA and IA. Furthermore, an ITA, as defined in this study, is indicative of the synergy between the ISA and IA. An ITA serves as the "pulling together" and extension of two often separate and disparate organizational architectures to form one overarching enterprise IT resource management architecture. Additionally, an ITA, depending on its maturity, expands the scope of the previously mentioned organizational architectures to include resources and relationships external to the organization. Thus, at the heart of the consolidation of the ISA and IA into the ITA still lies the issue of the maturity of the ITA. Just as multiple architectural representations and foundations exist, there are variations in the representation of the ITA. Gibson's (1994) testing and validation of four architectural types and Ross's identification of four distinct stages of architectural maturity appears to lend credibility to this notion. Gibson's (1994) research of ITA centered on the empirical testing and validation of four generic architectural types. He explicated that the four generic types would mature over a period and that they would eventually demonstrate some strategic impact on the firm. Each type of architectural types identified by Gibson (1994) had a greater or similar level of maturity. However, these generic architectural types were not treated as linear growth stages; rather, Gibson (1994) suggested that a single architectural type is most appropriate for a given situation.

Ross (2003) took a learning-in-stages approach to discuss ITA. She concluded that there were four stages of architectural maturity – the application silo stage, the standardized technology stage, the rationalized data stage, and the modular stage (see Figure 2.1). Firms in the application silo stage focus their resources on developing individual applications. Firms in the standardized technology stage focus their efforts and resources on the development of a shared infrastructure. Firms in the rationalized data stage, focus their efforts and resources on data management and infrastructure development. Firms in the modular stage focus efforts and resources on attaining strategic agility.

Ross (2003) found that for firms to develop and implement architectures representative of the various stages, they were required to have organizational competencies in ITA. She suggested that organizational competencies are needed to develop synergy between business strategy and ITA. She outlined strategic implications of IT associated with ITA maturity, and revealed IT capabilities a firm can derive from its ITA (Figure. 2.1), depending on its maturity.

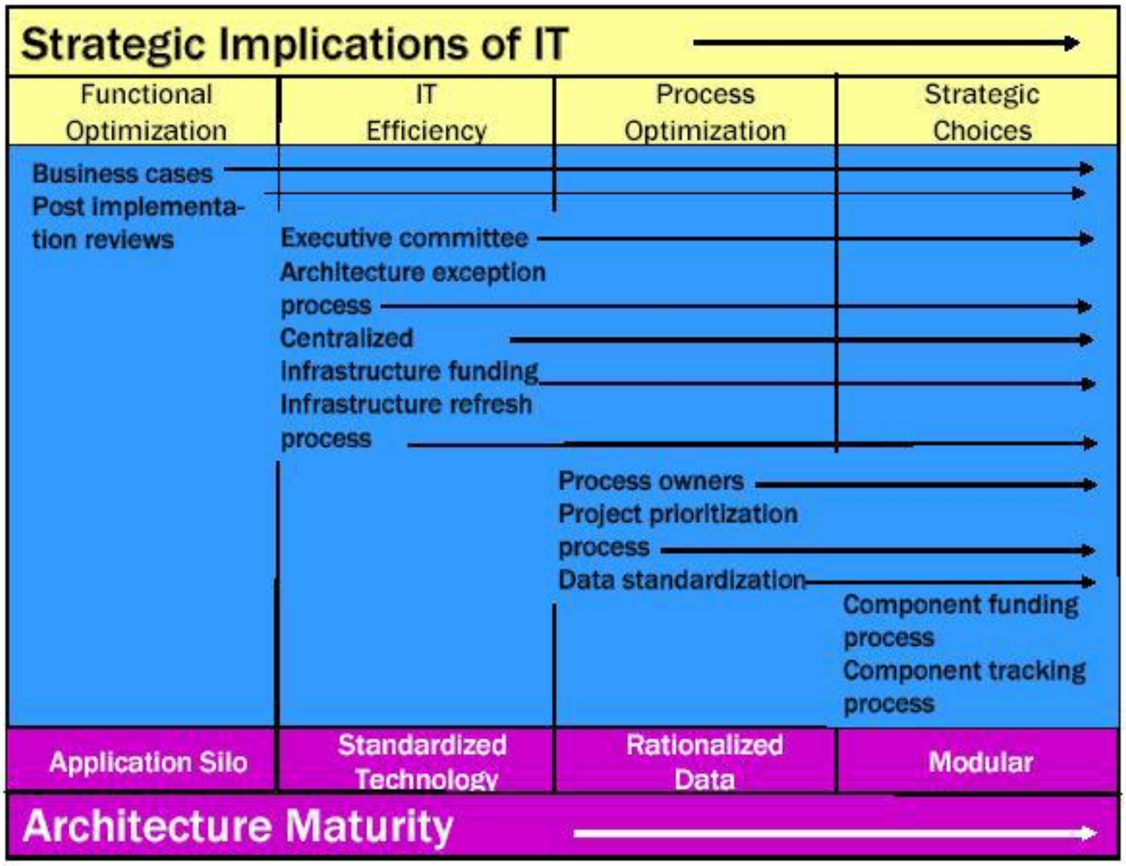


Figure 2.1 ITA Maturity Model (Ross, 2003)

2.3. Strategic Alignment

The challenges of aligning the IT function with the rest of the business have been highlighted in a number of studies (Broadbent & Weill, 1993; Chan, 2002). The numerous definitions of strategic alignment have been a major impetus towards the further proliferation of the challenges of attaining it. For instance, Reich & Benbasat (1996) defined alignment as the degree to which the IT mission, objectives, and plans are supported by the business mission, objectives, and plans. Others have defined strategic alignment somewhat differently. Henderson & Venkatraman (1993) defined strategic alignment as the fit between an organization and its strategy, structure, processes, technology and environment (see Figure 2.2). Chan et al., (1997) defined strategic alignment as the fit between business strategic orientation and IS strategic orientation. Furthermore, Chan et al. (1997) argued that alignment is a state or outcome. Additionally, Reich and Benbasat (2000) stated that processes, communication, and planning are determinants of alignment.

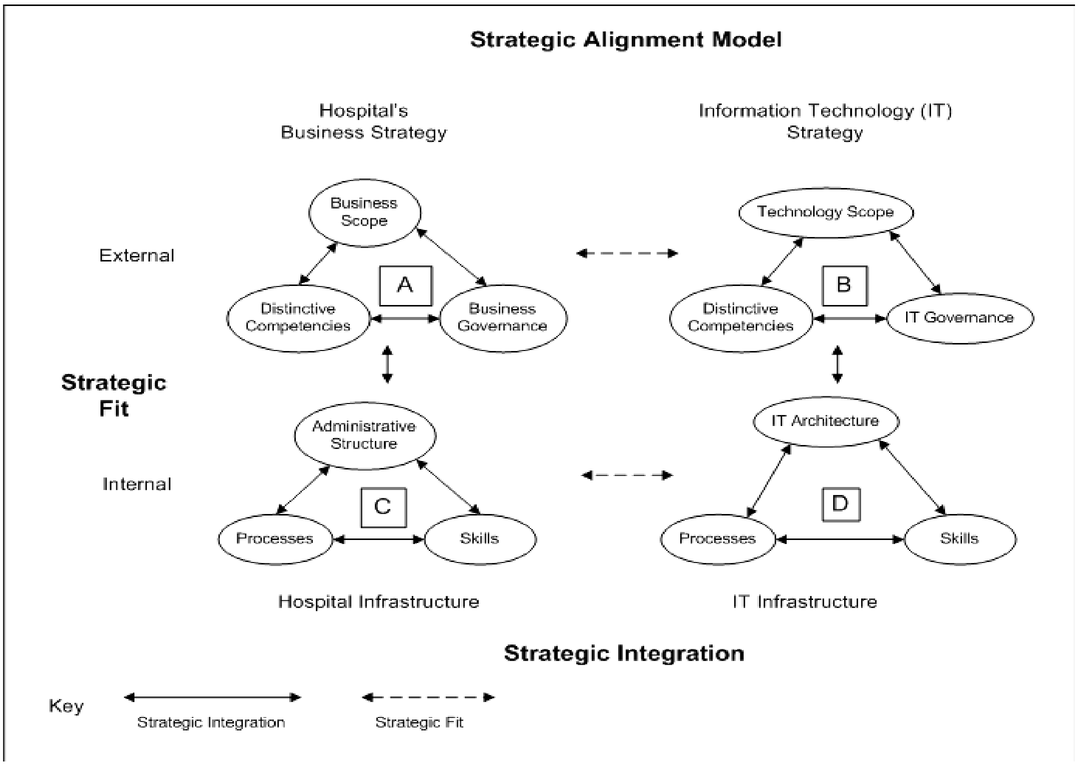


Figure 2.2 The Strategic Alignment Model (Ross et al., 2003)

Many firms have been faced with attempting to align IT and business strategies. The challenges of such alignment have been highlighted in a number of studies (Chan, 2002; Luftman, 2000). However, regardless of the challenges, many firms endeavor to attain strategic alignment. This is partly due to the ever-increasing benefits that are believed to be awaiting those firms that are able to balance their IT and business functions.

2.4. Enterprise Systems Classification models

The concept of application classification has been known since the first application was launched and has been exploited by several authors (eg. Anthony, 1965; Gorry & Scott, 1971; Leek ,1997; Ward &, Peppard 2002). As other classification models, like Boston matrix developed by Boston Consulting Group, application classifications help to understand the field in a structured manner. Anthony introduced the first hierarchical application portfolio in 1965. He classified types of management activity (Anthony, 1965). Gibson and Nolan (1974) and (Nolan, 1979) introduced the first model of evolution of information systems in an organisation. This was based on Anthony's classification of management activity. These models provided the first framework for application classification. The model by Anthony is illustrated in Figure 2.3.

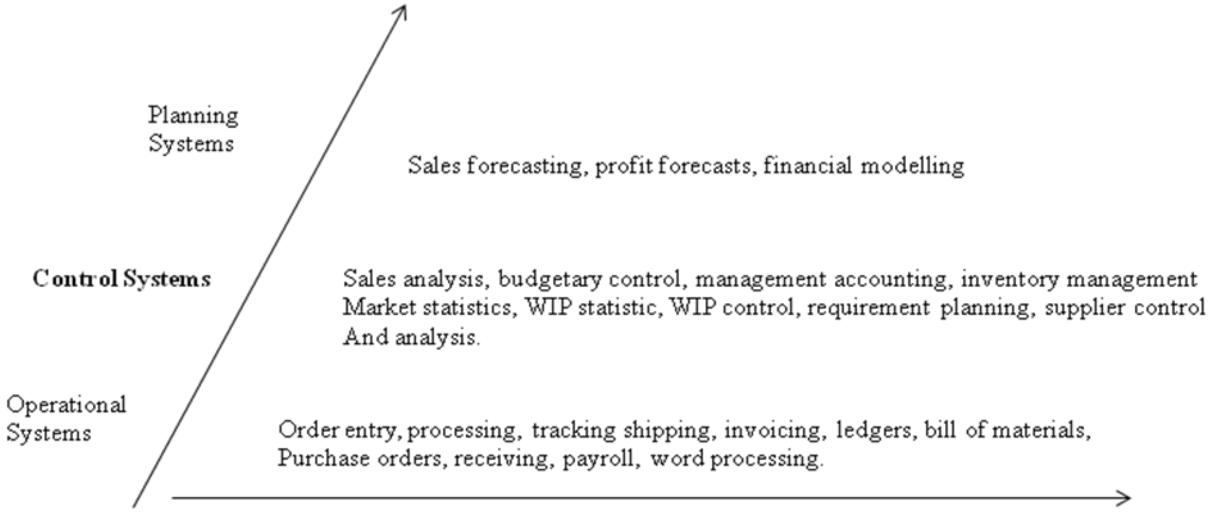


Figure 2.3 Classification of planning, control and operational systems (Anthony, 1965)

In Anthony's model, the classification is based on stratification of management activity in an organisation. Anthony divided management activity into strategic planning, management control and operational control. He used the same logic in classifying applications because they were built to support different levels of management activity.

Gorry and Scott (1971) suggested that management information system should be looked at from a decision-making point of view. The reason was that management information system should exist only to support decisions. To help understand management information systems, Gorry and Scott created a Management Information Systems framework. This framework was based on the work of Anthony and Simon (Anthony, 1965; Simon, 1960) on managerial activity. In their framework, Gorry and Scott have combined Anthony's classification of managerial activity and Simon's classification of the types of decision problems a manager has to deal with (structured, semi-structured and unstructured). Then they have classified management information systems in this matrix. The framework by Gorry and Scott is illustrated in Figure 2.4

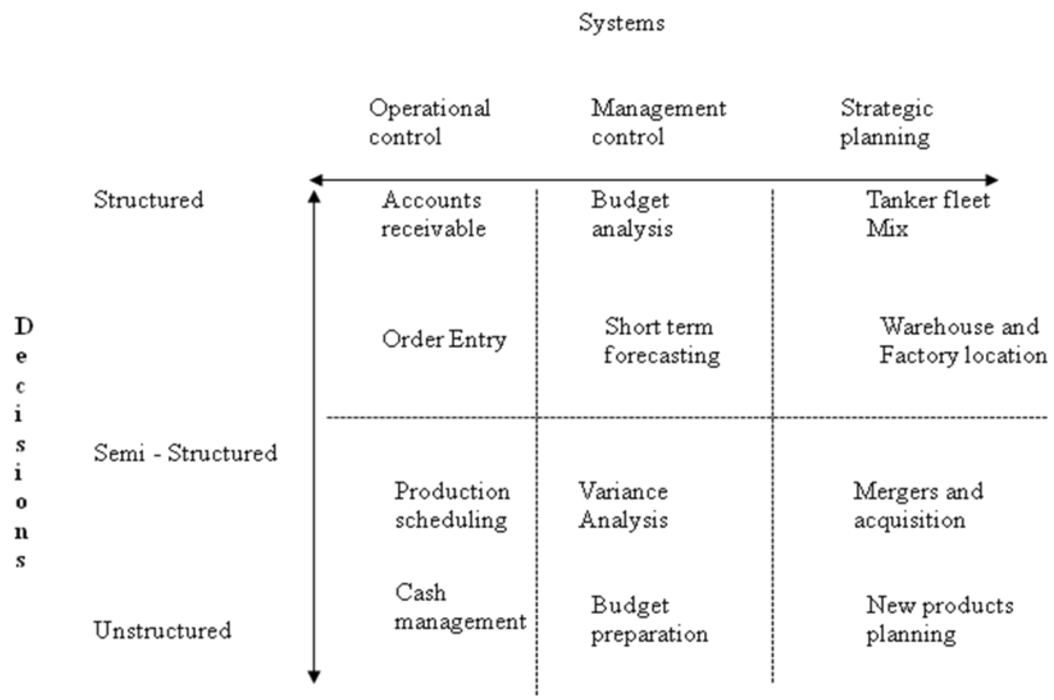


Figure 2.4 Information System Framework (Gorry & Scott, 1971)

The classification matrix by Gorry and Scott contains a class of decisions that are called semi-structured. They defined semi-structured decisions as decisions where one or two of the three phases intelligence, design and choice, are unstructured. In the framework shown in Figure 2.4, decisions above the dotted line are largely structured and decisions below the line are mainly unstructured. Gorry and Scott named decisions supporting information systems as "Decision Support Systems". In Gorry and Scott's framework, there are six cells but they stressed that these cells are not well defined categories.

Information systems have changed considerably since the framework by Gorry and Scott was published, but it can still be argued that the basic concept is valid. Utilization of information systems has become a core of every company's operative processes. Almost all activities that Gorry and Scott called structured are now managed through information systems. In perspective to Gorry and Scott's framework, the future trend in utilizing application systems seems to be towards decisions and strategic planning.
During the first two decades of the time of information systems, the 1960's and the 1970's, they were mainly exploited for improving operational efficiency (data processing) and for improving management effectiveness by satisfying information requirements for decision making (management information system). In the early 1980's

began the time of strategic information systems (Wiseman, 1985; Galliers & Somogyi, 1987; Friedman, 1994). Ward & Peppard (2002) defined a strategic information system as a system that improves competitiveness by changing the nature of conduct of the business.

Benjamin et al. (1984) classified types of potential opportunity where strategic information systems can help, to those that focus on the market place and to those that focus on internal operations. In internal operations IS/IT can be used to improve traditional ways of business or to make 'significant structural changes' in the way a company does business. Notowidigbo (1984) divided strategic information systems into internal systems that benefit the company directly and external systems that benefit the company's customers.

Venkatraman (1991) evaluated the strategic benefits that have been achieved by using information systems that changed the business. He identified three types of use of information systems. 1. Business Process redesign – using IS/IT to realign business activities and their relationship to achieve performance breakthrough. 2. Business Network redesign – changing the way information is used by the organisation and its trading partners thereby changing how the industry overall carries out the value-adding process. 3. Business scope redefinitions – extending the market or product set based on information or changing the role of the organisation in the industry.

The application architecture of a company needs to be planned and managed. According to McFarlan (1984) this should be based on current and expected future contributions of application to the company. McFarlan (1984) presented an application classification model that classifies applications in four classes based on their business contribution. Mcfarlan's grid is illustrated in Figure 2.5.

Strategic	**High Potential**
Application that are critical to sustaining future business strategy	Applications that may be important in achieving future success.
Applications on which the organisation currently depends for success	Applications that are valuables but not critical to success
Key Operational	support

Figure 2.5 Applications portfolio (McFarlan, 1984)

The model presented by McFarlan (1984) suggested analysis of current, planned and potential applications and to divide them into four categories based on their current and future importance to the business. The categories are (categories names were modified by Ward 1990, original names are in brackets).

i. Strategic (Strategic) applications are critical to future business success. They create or support changes in how the organisation conducts its business with the aim of providing competitive advantages. If the technology used is 'leading edge' technology or not does not indicate if the application is strategic or not.

ii. Key operational (Factory) applications sustain exiting business operations, helping to avoid any disadvantage. These kinds of applications are for example ERP or Point of Sale systems.

iii. Support (Support) applications improve business efficiency and management effectiveness but do not in themselves sustain business or provide any competitive advantage.

iv. High potential (Turnaround) innovative applications may create opportunities to gain future advantage but this has not been proved yet.

Applications can move around the matrix during their lifetime. This may be for instance, because innovative applications are not innovative anymore after a certain time period but becomes key operational applications. Equally, as a general trend, it is typical that applications which were for instance of high potential in the 1980's are not necessarily high potential anymore in the 21st century.

Originally this classification was intended to plot the overall expected contribution of information systems to business success in the enterprise. In practice, this is not useful in most cases because most probably an enterprise has applications in all the categories and there will be changes in the application portfolio overtime. These limitations of the original McFarlan grid have been described by Hirschheim et al (1988), who found that it was an unhelpful way of categorizing the whole set of information systems since practically every company had systems in all four categories. Still McFarlan grid is very useful when classifying applications within the enterprise for the purpose of managing current and future application portfolios and setting priorities for investments (Ward & Peppard, 2002).

Ward and Peppard (2002) suggested an analysis to be performed with the following deliverables when evaluating a current application portfolio:

i. Categorization in terms of application portfolio segments: strategic, high potential, key operational and support.
ii. Assessment of coverage and contribution of systems to business needs and any major opportunities to increase business value.
iii. The extent to which systems integrate or interoperate.
iv. Assessment of the effectiveness and robustness and the unrealized potential in current systems and of the enhancement required to increase contribution.
v. Common elements and differences between current portfolio and required information and system architecture.
vi. Supporting information to allow prioritization of enhancement and support work on current systems.
vii. Opportunities that exist to improve information quality.
viii. Strengths and weaknesses assessed against the business critical success factors.
ix. Assessment of the risks of failure from current portfolio.

The information technology assessment and adoption (ITAA) matrix developed by Munro and Huff (1985) is based on work by Benjamin et al., (1984) and considered how organisations have adopted IT/IS as a competitive weapon. Most companies, according to Munro and Huff (1985), are technology driven and are looking for ways to deploy new technology to advantage or issue driven looking for new business opportunities within the known possibilities of existing technology.

Gallier (1987) developed a matrix that considers factors affecting the planning methods, but this time in relation to long term and short term thinking, strategy or issue driven and business issues versus technology driven planning. He separates the need for IS/IT to react to current business issues (key operational) from the need to react to changing future objectives (strategic) and compares them to the proactive IS/IT required for high potential opportunities. Like many others, he recognizes the importance of managing support systems in an efficient way.

Turban et al. (2001) introduced a classification of information systems based on the number of users and the level of decision making. They classified information systems into six groups. As a seventh category, they introduced an information network infrastructure in their model. The Turban et al.(2001) model is illustrated in Figure 2.6.

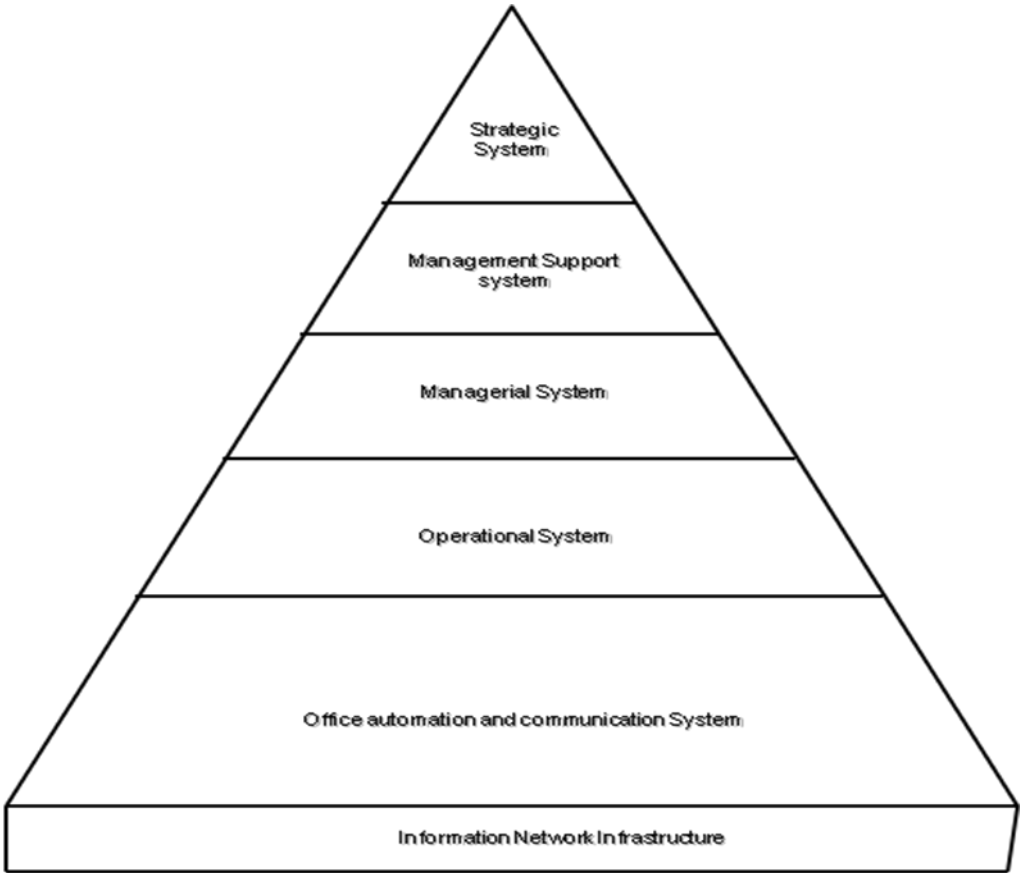


Figure 2.6 Classification of information systems based on the number of users and level of decision-making (Turban et al., 2001).

The systems groups in Turban et al. (2001) classification are:

i. Office automation and communication systems are used by all employees. Examples of these systems are word processing, spreadsheets and e-mail applications.

ii. Operational systems support organisation in everyday actions such as registration of time used for different tasks, making of a purchasing order and the making of a travel expense report. The users are employees and line management. According to Turban et al. operational systems include also systems for transaction processing, management information systems and support systems for decision-making.

iii. Managerial systems are for planning, organizing and controlling in the short run. This group includes systems for statistical abstracts, reports of divergences, periodical and ad hoc reports, comparative analysis, and projections like trend analysis, sales in the future and cash flow. The users of management systems belong to the middle management.

iv. Management and Expert support systems provide support information and knowledge creation and their integration to business. Search engines in the internet, computer aided design (CAD) and Computer integrated manufacturing (CIM) are examples of this group. The users are experts like economists, lawyers and marketing experts.

v. Strategic systems support decisions that significantly change the way we do business. Strategic systems are both reactive and proactive. The users of strategic systems belong to the top management.

vi. Information network infrastructure includes computers, networks, databases and other devices.

There are some issues that can be criticized in the model by Turban et al, (2001). First, it is difficult to see the connection between the level of organisations and the systems as they present it. The most peculiar claim is to connect employees with office automation and communication systems. It seems obvious that every level of organisation uses these systems and not just employees. Another interpretation of the model would be that in addition to employees, all levels of organisation shown above the employees in the model use office automation and communication systems and on the other hand, all other levels except employees use operational systems. However, this logic is not valid, because for instance operational systems are not used by top management. Altogether, the model from Turban et al. does not seem logical.

Weill and Broadbent (1998) published their information technology portfolio in 1998. Their classification model starts out with four different management objectives in assessing information technology investments. These four objectives are supported by four class of information technology. It should be noted that the classification by Weill and Broadbent includes technologies and not just applications. The four classes in the model are IT infrastructure, transactional infrastructure, informational infrastructure and the strategic IT. The Classifications by Weill and Broadbent is illustrated in Figure 2.7.

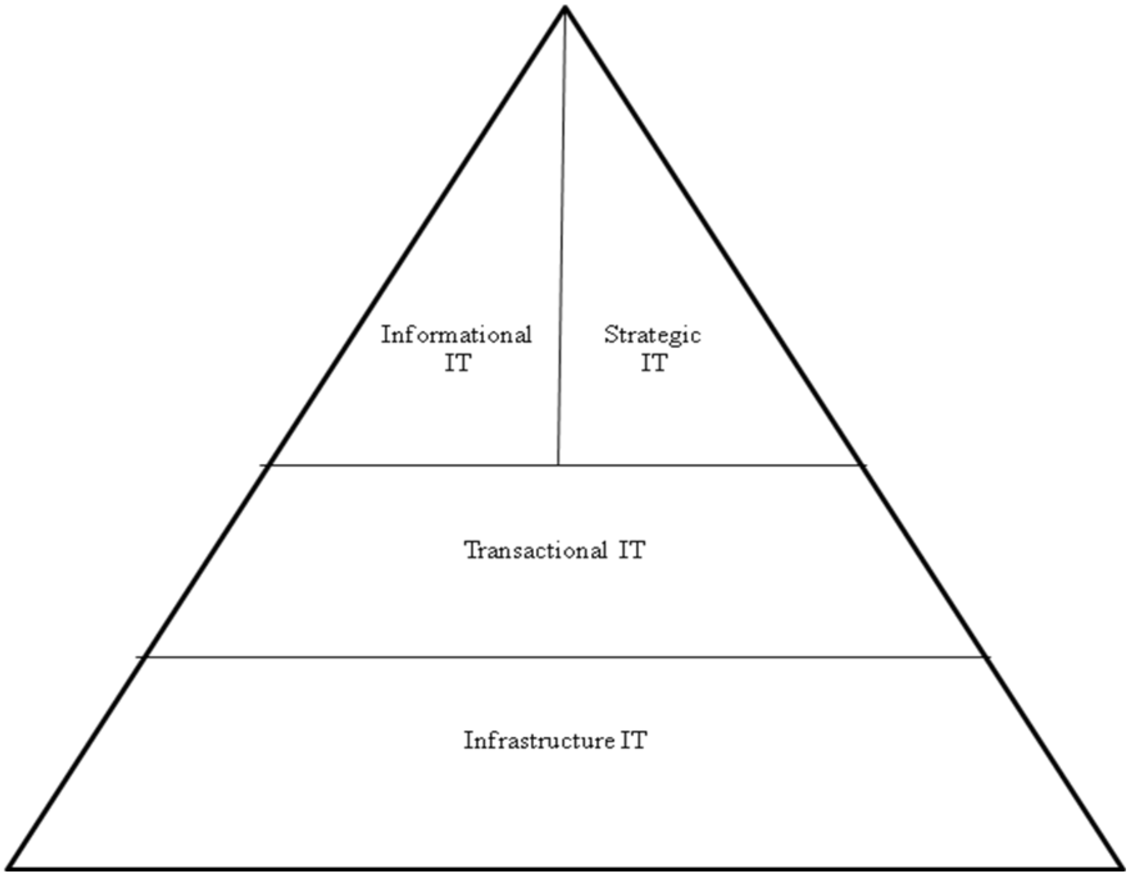


Figure 2.7. IT Classification model (Weill & Broadbent, 1998).

IT infrastructure is the base for applications in an enterprise. The technical infrastructure is a shared facility across the enterprise to run multiple applications.

Transactional IT processes the basic transactional processes in the company aspiring at cost reduction. This category includes systems like payroll and order processing.

Informational IT systems collect and refine information for enterprise management. Planning, communication, decision-making, accounting and knowledge management are typical areas of these systems. Information is typically collected from several transactional systems.

Strategic IT is a category equivalent to the strategic category in McFarlan's model, the aim is to gain competitive advantage or major innovation. The strategic potential of the technology is typically limited in time and systems or technologies move from this category to another one during their life cycle.

Organizations are always influenced by the internal environment where they operate. This context is likely to be different in different companies and it has impact on information systems.

Sullivan (1985) proposed a simple matrix to analyze how forces outside the control of an individual part of the organisation affect the IS/IT strategic environment. He has two axes in his matrix (Sullivan, 1985):

i. Infusion - the degree to which a company becomes dependent on IS/IT to carry out its core operations and manage the business.

ii. Diffusion – the degree to which IT has become dispersed throughout the company and decisions concerning its use are devolved.

These axes not only reflect the increasingly strategic nature of IS/IT, but also the changing economics of technology and the ability to use it without the need for skilled technical staff (Ward & Peppard, 2002). Sullivan's matrix is illustrated in Figure 2.8.

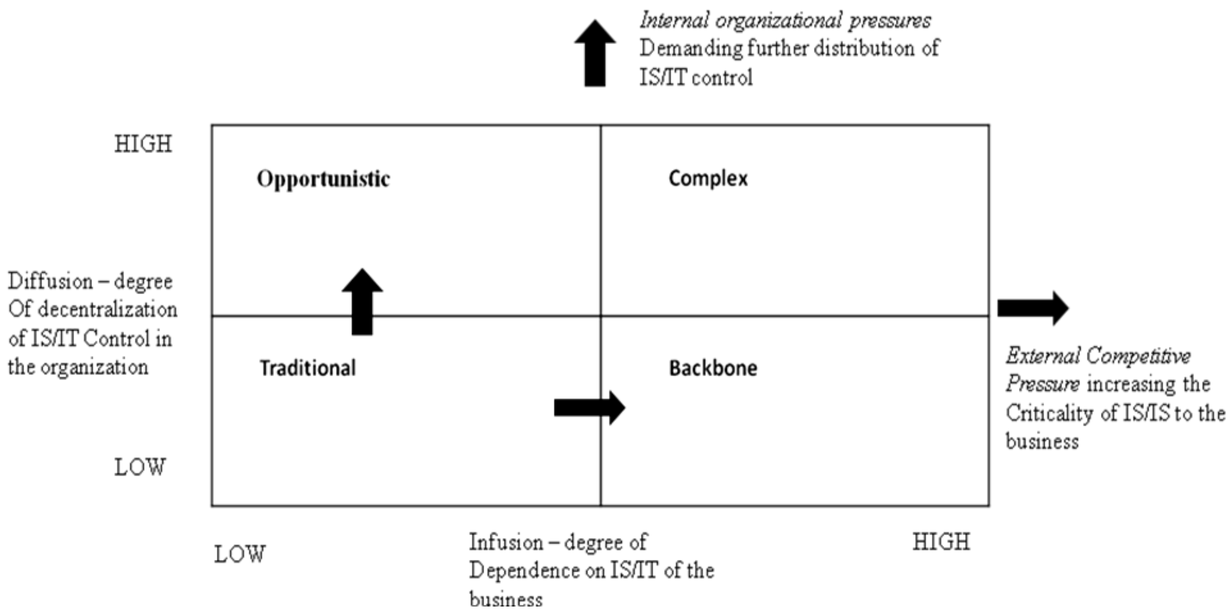


Figure 2.8. Sullivan's matrix of environment of IS/IT strategy (Sullivan, 1985)

By marking high and low degrees of infusion and diffusion, four different environments are established. These are:

i. **Low diffusion/low infusion** – highly centralized control of IT resources while IS is not critical to the business. This is described as a 'traditional environment, typical of companies using IT solely to improve efficiency in a system-to-system basis.

ii. **Low diffusion/high infusion** – highly centralized control where IS is critical to business operations and control. The business could be seriously disadvantaged if systems fail. Therefore, high-quality systems with a high degree of integration are required.

iii. **High diffusion/low infusion** – largely decentralized control, giving business managers the ability to satisfy their local priorities. Any integration of systems occurs due to user-user co-operation, not due to overall business or IT design. The management approach is 'opportunistic', driven by short-term priorities that may create business advantages in some areas.

iv. **High diffusion/high infusion** – largely decentralized control but the business depends on the systems for success, both in avoiding disadvantages and in achieving its overall business objectives. Sullivan describes this as a 'complex' environment that is difficult to manage. Too much central control to avoid poor investments will limit innovations hence new strategic opportunities may be compromised; too little control and the core systems may disintegrate.

Kalakota and Robinson (2000) introduced an e-business architecture, which is an integrated application framework. When comparing Kalakota and Robinson's framework to other applications classification models, the main difference is that their model covers external parties too. This is a consequence of their emphasis on e-business. In addition to e-business, Kalakota and Robinson stressed the importance of integrating all the parts of an application architecture. In their model, applications are classified based on their purpose. According to Kalakota and Robinson (2000) their framework can assist management to grasp the big picture and to set priorities in building e-business enterprise. The classification model by Kalakota and Robinson is illustrated in Figure 2.9.

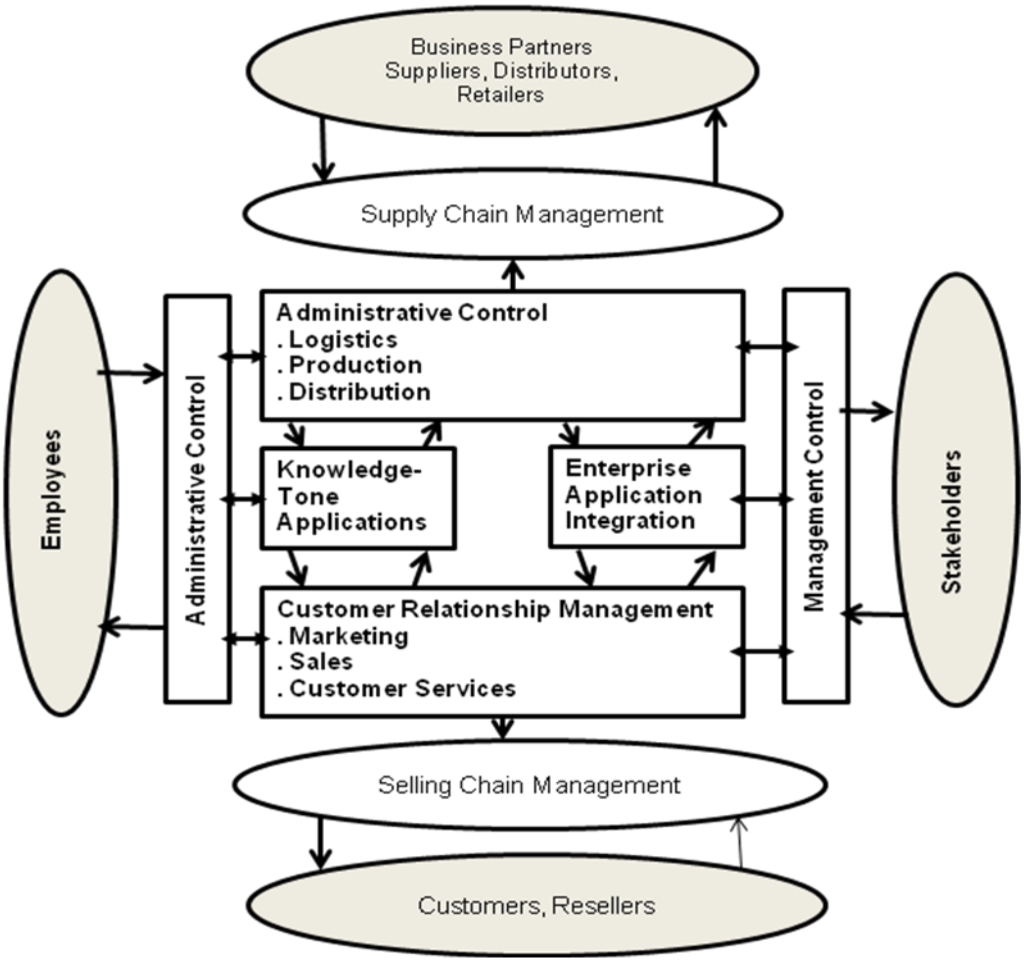


Figure 2.9. Information System architecture for e-business (Kalakota & Robinson, 2000).

As shown in Figure 2.9, Kalakota and Robinson's model has several application classes. They are ERP, Customer Relationship Management, Business Intelligence, Enterprise Application Integration, Supply Chain Management, Selling Chain Management, Management Control and Administrative control. Application classes are relatively small and quite different compared to other application models. As described in Figure 2.9, there are many ways of classifying applications and over the years, several models have been introduced. Most of the classification models have good features and it's a matter of one's personal opinion which model is better that the other. Feasibility of different models depends on the purpose. Therefore, attention should be paid to choosing the right model before using any of them.

2.5 Enterprise Application Evolution Models

One perspective in analyzing the development of information systems and their use in organisations is to look at them from the evolution point of view. Gibson and Nolan (1974) introduced the first evolution model in 1974. They developed an evolution model which illustrated four stages of growth (Gibson & Nolan, 1974) and later Nolan (1979) added two further stages. Nolan and Gibson's model is illustrated in Figure 2.10.

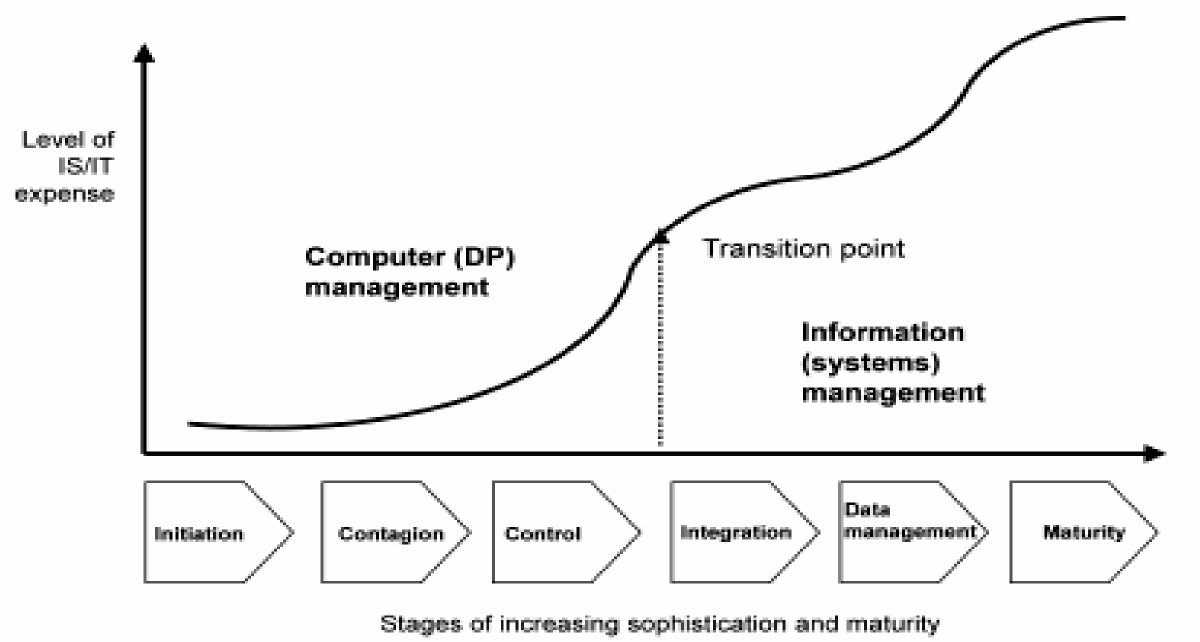


Figure 2.10 Stages of evolution of IS/IT in relation to expenditure (Gibson & Nolan, 1974).

Stages in the model by Gibson and Nolan are as follows:

i. Initiation: mainly batch processing to automate clerical operations. The focus is on operational systems and thus cost reduction.

ii. Contagion: IS/IT expenses rise rapidly, because users demand more applications and have high expectations of benefits. There is not much control.

iii. Control: Management is concerned about increased cost and takes IS/IT under control. As a result, users are often dissatisfied because the project delays and there is a high level of control.

iv. Integration: there is considerable expenditure on integrating existing systems. Data processing provides service to users and not just solution to problems

v. Data administration: information requirements rather than processing drive the applications portfolio and information is shared within the organisation.

vi. Maturity: business development and planning and development of IS/IT in the organisation are closely co-ordinated.

Nolan’s model was criticized by Benbasat et al., (1984) and King and Kramer (1984) because they claim that empirical support for the model is generally weak. Drury (1983) noted that particularly in later stages of the model the real world is much more complex than what is reflected in the model. Despite the criticism, the model continues to be quite commonly used among practitioners (Ward & Peppard, 2002). According to Friedman (1994) the introduction of strategic information system fundamentally changed the concept of how IS/IT evolves to maturity in Nolan’s model.

According to Kalakota and Robinson (2000), evolution of business application has been a process of three stages as listed below;

i) In the first stage, task-oriented applications, was when business applications were narrowly focused and task oriented, simplifying processes such as order entry. Although task specialization improved productivity of a single task, the disadvantage was fragmentation of processes.

ii) In the second stage, the aim was to build functional applications, when task oriented applications became more functionally integrated, for instance, order entry was integrated with sales application. Kalakota and Robinson (2000) called the third phase a stage of integrated cross-functional applications.

iii) In the third stage, applications are integrated cross-functionally and they support seamless process flow across various functions. ERP and CRM applications are examples of integrated cross-functional applications.

Porter (2001) acknowledged very similar stages of evolution in business as Kalakota and Robinson. Furthermore, Porter defines a fourth and fifth stage. According to Porter, a fourth stage enables the integration of the value chain and entire value system, that is, the set of value chains in an entire industry, encompassing those of tiers of suppliers, channels and customers. Porter continued that in a fifth stage, information technology will be used not only to connect the various activities and players in the value system but to optimize its workings in real time. For instance, choices will be made based on information activities and corporate entities.

Riihimaa and Ruohonen (2002) studied the interaction between management problems and responses provided by information and telecommunication technology. They presented several stages of development in management themes and emphasis on information systems development to correspond with those challenges. The stages presented by Riihimaa and Ruohonen are shown in Figure 2.11.

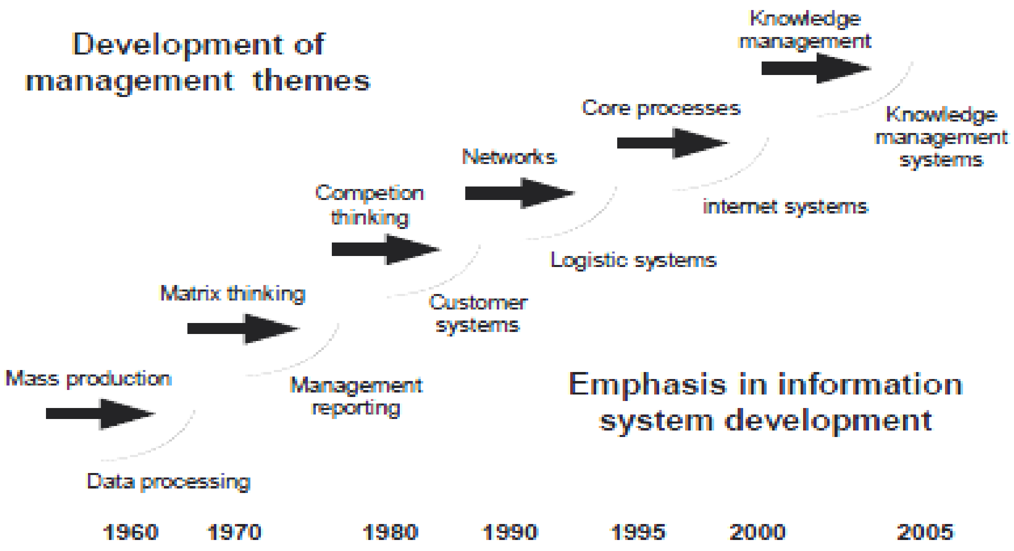


Figure 2.11. Development of Management themes and emphasis on information system development (Ruohonen, 2002).

Riihimaa and Ruohonen (2002) studied the development of information systems usage in Finnish industrial companies. They describe four different forces that drive evolution and change towards e-business. These are business development, technology development, organizational maturity and organizational potential to utilize e-business (Riihimaa & Ruohonen, 2002):

i. **Business development** – A rapidly changing business environment defines the possibilities for success for many companies. Consolidation and other changes in business sector impact organisations. The change from a product oriented business to a more service oriented business guides the way to operate in a company.

ii. **Technology development** – depending on the company and industry sector technology may be the driving force in the development of the industry. This is the case in some areas of information technology, information systems and telecommunication. For example, in office automation, Microsoft has established a strong market position for their product and the same applies to SAP in ERP systems

iii. **Organizational maturity** – business environment set the frame, but the real change towards e-business is defined by the organisation’s own ability to learn and adjust its processes to the demands set by customers and partners.

iv. **Organizational potential to utilize e-business** – In some business sectors, such as banking and insurance, utilizing information and communication technology is a natural part of the business. For more traditional business sectors utilizing information technologies is not that obvious, but it is used there too. Good examples are supply chain management in the automotive industry or customer service and integration solutions in the engineering industry. A company’s potential to utilize new technologies is not limited to automating processes; it can also mean introducing a new service or product or process innovation.

Riihimaa and Ruohonen (2002) made a survey of the technology industry in Finland concerning development of information systems in organisations. They identified four stages of development of information systems in organisations. According to their survey, these phases also apply on the order of development within a company. The development phases are:

i. ERP-Phase. Building and integrating basic systems within the company.

ii. SCM-phase (Supply Chain Management). The focus is in aiming at a more efficient supply chain within the company or with the network of companies. Information systems play an important role and support this effort.

iii. CRM-phase (Customer Relationship Management). The focus is to create value add services for customers. The ability to collect and manage customer related information is essential. The supply chain has to operate properly before customer information can be effectively utilized. Particularly sales and order fulfillment processes need to be under control and systems need to be integrated.

iv. Knowledge management. The objective in the knowledge management phase is to create partnerships and share common knowledge within the company network. Characteristics for this phase are innovative ways to produce, distribute and develop products. In some cases, services become an inseparable part of the products.

Application evolution models are useful instruments in analyzing the development of applications in an organisation. Evolution models can be used for evaluating the general development level of applications in the organisation. Equally, application evolution models are practical in analyzing the order of implementation of different applications within an organization.

2.6 Small and Medium - Sized Enterprises

Small and medium-sized enterprises are vibrant and growing sector in most economies around the world (Levy & Powell, 2005). There is no doubt as to the importance of SMEs, as they are an important driver of the economy and the major source of employment.

In Europe, they represent 99.8% of the total number of enterprises, and provide around 75 million jobs, or 70% of the total employment.(European Commission, 2011)

The European commission defined an SME as an enterprise having less than 250 employees, a turnover of less than Ɛ50M or a balance sheet total of less than Ɛ43M. (European Commission, 2003).

The Central bank of Nigeria defined a Small and Medium Scale Enterprise (SME) as an enterprise that has asset base (excluding land) of between N5million –N500 million and labour force of between 11 and 300 (CBN, 2001).

Levy and Powell (2005) indicated that SMEs are not a homogeneous group, but are heterogeneous with diverse needs and objectives. When developing IS techniques, different approaches are needed for enterprises with different characteristics (Levy & Powell, 2005). As SMEs are the selected target group in this study, it is necessary to understand their characteristics and drivers of growth. Bernaert et al. (2013) mentioned characteristics on which SMEs can be distinguished, such as growth, whether the company is family owned or not, influence of the CEO and education of the CEO.

2.6.1 Information Technology Adoption in SMEs

Although IS and IT in large firms provide major opportunities for added value through exploitation of information, there is less evidence of SMEs investing in these matters to capture similar benefits. (Levy et al., 2002) This weaker degree of IT adoption indicates missed opportunities, as the exploitation of information and knowledge is documented as one of the modern times key factors to competitive success for SMEs (Lybaert, 1998).

It is understandable though, as IT adoption is in general more challenging for small firms. Specific challenges are formed by their small size and unsophisticated deployment of information systems (Bidan et al., 2012) The difficulties faced are caused by a lack of technical competence and know-how, shortages of human resources, risk-aversion, insufficient support technologies, or organizational structures (Duhan et al., 2001, Themistocleous and Chen, 2004).

From previously mentioned and other SME characteristics found in literature studies, Bernaert et al. (2013) listed six well-documented SME characteristics that influence their IT adoption:

i. Employees and management are constrained by time, leaving little time to look at strategic matters.

ii. SMEs have limited IT knowledge and technical skills.

iii. SMEs have fewer resources. The smaller the company, the fewer resources it has, for instance to hire experts.

iv. There is a big demand in SMEs for knowledge regarding the performance of tasks and how things are done.

v. The manager or CEO, who is often the company owner, is commonly the single person who decides on strategic issues.

vi. In order to accept a new IS approach, the expected returns must exceed the expected risks and costs.

It is clear that these SME-specific characteristics differ from those of larger companies, and thus an adapted IT approach is preferred.

2.6.2 SME Challenges

Enterprises have to be flexible and be able to adapt in response to a changing competitive environment. Globalization, the evolving roles of IT in enterprises and compliance regulations are just a few of the many challenges that force enterprises to adapt in order to survive (Proper and Greefhorst, 2011). In response to these challenges, the enterprise, large or small, thus evolves continuously. As a result, its structure changes accordingly.

Without a certain means of control, structural changes can lead to an overly complex and uncoordinated environment that is hard to manage (Proper and Greefhorst, 2011).

A good example is growth-oriented enterprises. Growth is one of the most common objectives of SMEs, though growth may even present itself without being planned. Among other, growth-oriented companies enter new markets, hire additional personnel, increase the number or scope of their processes and expand their applications and technology infrastructure to support them. The focus on its core objectives, the ability to provide clear personnel job descriptions and asset management can soon be lost. Therefore, companies with a high growth rate need a clear vision providing the company's strategic direction (Levy & Powell, 2005).

2.7. Enterprise Architecture

Enterprise Architecture (EA) terminology carries many variations within each organization and in the vast array of literature. Therefore, the authors have settled on one consistent set of definitions for key terms used within this guide. The definition for *Enterprise Architecture* is the endorsed definition from the Federal Chief Information Officer (CIO) Council and appears in the September 1999 version of the FEAF. Although the term *enterprise* is defined in terms of an organization, it must be understood that in many cases, the enterprise may transcend established organizational boundaries (e.g., trade, grant management, financial management, and logistics).

Enterprise Architecture is defined as a strategic information asset base, which defines the mission, the information necessary to perform the mission and the technologies necessary to perform the mission, and the transitional processes for implementing new technologies in response to the changing mission needs. An enterprise architecture includes a baseline architecture, target architecture, and a sequencing plan.

The primary purpose of an EA is to *inform, guide,* and *constrain* the decisions for the enterprise, especially those related to IT investments. The true challenge of enterprise engineering is to maintain the architecture as a primary authoritative resource for enterprise IT planning. This goal is not met via enforced policy, but by the value and utility of the information provided by the EA.

2.7.1 Enterprise Architecture Frameworks

Common understanding of the enterprise and the business are the cornerstones in developing any information system. Enterprise architecture models help creates this common understanding. As defined earlier, enterprise architecture describes structures and processes within an enterprise and consists of four elements. These elements are the business (or business process) architecture, the application architecture, the information architecture and the technology architecture. Enterprise architecture is a tool to help executive think about the organization as a whole (Harmon, 2003).

The primary reason for developing enterprise architecture is to support the business by providing the fundamental technology and process structure for an IT strategy (The Open Group, 2002). Enterprise architecture can by very complex and the integration of all the enterprise architecture components is essential for the organization to easily evolve and successfully adapt to the frequent technology and business change that occur. An enterprises architecture framework can aid development of adaptable enterprise architecture (Morganwalp & Sage, 2003)

Morganwalp (2003) listed desired feature for an enterprise architecture framework:

- encompasses enterprise - wide views - Since an enterprise architecture affects the whole organization, different stakeholder views should be include

- Facilitates system integration – An organization has to integrate its business processes with internal as well as with external applications. Therefore it is essential that the enterprise architecture framework supports and facilitates this integration challenge.
- Driven by or founded business requirements – As any other information system planning, an enterprise architecture framework has to be based on business requirements.
- Flexible enough to support changes in technology and or business – The business and technology environment is changing continuously. Therefore the enterprise architecture framework has to be flexible and agile to react to these changes.

Schekkerman (2003b) listed similar critical success factors for an enterprise architecture framework:

- Holistic in scope: It must address all aspects of the extended enterprise and be directly associated with the business technology alignment: business structure, business activities, business processes, information flows, information – systems infrastructure, standards and policies.
- Alignment driven: It must address the need to directly align extended business and technology drivers in a way that is comprehensible and transparent to all key stakeholders, with a continued process of tracing enterprise architecture initiatives of the business strategy.
- Value driven: It must provide mechanisms to define business cases that help ensure and demonstrate the business value of enterprise architecture solutions.
- Dynamic environments: it must include analytical methods that support the development of enterprise architectures that are flexible and dynamic to changing business drivers and new opportunities.
- Normative results: It must provide the ability to define solution sets that can be measured, validated and mapped to real world solutions.
- Non prescriptive: It must not presume an implementation approach, because that would be out of the scope of the enterprise architecture program.

Morganwalp (2003) and Whitman et al., (1998) stressed the importance of multiple views in enterprise architecture and particularly that these views need to be integrated with each other.

Huff et al. (1993) argue that multiple perspectives of an enterprise are required depending on the purpose of the examination. They describe a five – point approach when studying an enterprise:

- Business Rule (or information) view defines the entities managed by the enterprise and the rules governing their relationship and interactions.
- Activity view defines the functions performed by the enterprise (what is done).
- Business process view defines time-sequenced set processes (how is it done).
- Resource view defines the resources and capabilities managed by the enterprise
- Organization view describes how the enterprise is organized. This includes the set of constraints and rules governing how it manages itself and its process.

All the enterprise architecture frameworks support a principle of having at least four separate perspectives business, information, information systems and technology (The Open Group, 2002; Zachman, 1987; Schekkerman, 2003a; CIO Council, 1999)

There are several enterprise architecture frameworks available. In this study some of them have been described and reviewed in more detail. The selected enterprise architecture frameworks are the Zachman framework, the CIMOSA, the ARIS, the TOGAF, the FEAF and the NCR enterprise architecture framework. These frameworks have be selected because they are cited most often in enterprise architectures literature and are recognized among practitioners using enterprise architecture framework. Excluded from this study include US DoD'S C4ISR (Command, control, communications, computers, intelligence, surveillance and Reconnaissance) (C4ISR, 1997), IBM's enterprise solutions structure (ESS) (Plachy & Hausler, 1999), Microsoft's Enterprise services Framework (Microsoft Corporation, 2003) and the Extended Enterprise Architecture (E2A) Frameworks were excluded because there are not much material available about them and there is no added value in repeatedly describing all of them.

According to Morganwalp and Sage (2003), quality and effectiveness measures for enterprise architectures are of much importance. Buchanan (2001) suggests that enterprise architecture quality should be measured against three dimensions: financial efficiencies, business effectiveness and architecture process.

2.7.2. Review of Major Architecture Frameworks and Reference Models

Enterprise Architectures are typically developed using some or all of the following components:

- **Reference Architectures:** Reference architecture is a document or set of documents to which a project manager or other interested party can refer for best practices.
- A **framework:** a framework is often a layered structure indicating what kind of programs can or should be built and how they would interrelate.
- A **methodology**: a system of methods used in a particular area of study or activity
- **Standards**: a level of quality or attainment. It can also be something used as a measure, norm, or model in comparative evaluations.
- **Modelling tools and languages:** Modelling tools are basically 'model-based testing tools' which actually generates test inputs or test cases from stored information about a particular model (e.g. a state diagram), so are classified as test design tools. A **modeling language** is any artificial **language** that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules. The rules are used for interpretation of the meaning of components in the structure.
- **Architecture Model:** Architecture Model aims to depict the abstraction of reality, which is consisted of individual components, their relationship and focus. Architecture Model including design rules and modeling procedure, process and technique are an important part of a framework for developing, structuring, defining languages and describing. Modeling techniques commonly used are entity relationship (ER) and UML (class diagram, use case). Meanwhile most models are graphic, however it does not necessary have to be graphic.

Enterprise architecture (EA) frameworks are giving particular attention in the literature and it is important to note the difference between enterprise architecture framework and methodology. An EA methodology provides a technique for capturing various aspect of a business and turning these into models while frameworks provide a structure within which these models can systematically be placed (The Open Group, 2002). Put another way, “Architecture frameworks are standards for the description of architectures”.

EA frameworks are useful constructs simply for the reason that, despite their apparent differences, most enterprises actually have a lot in common. The identification and codification of these common structural elements eliminates the necessity to develop new EA’s from scratch. Instead, a relevant framework can be adopted and adapted for use. Reference architectures take this logic one-step further by tailoring an EA framework to a specific industry, perhaps with some models already partially developed.

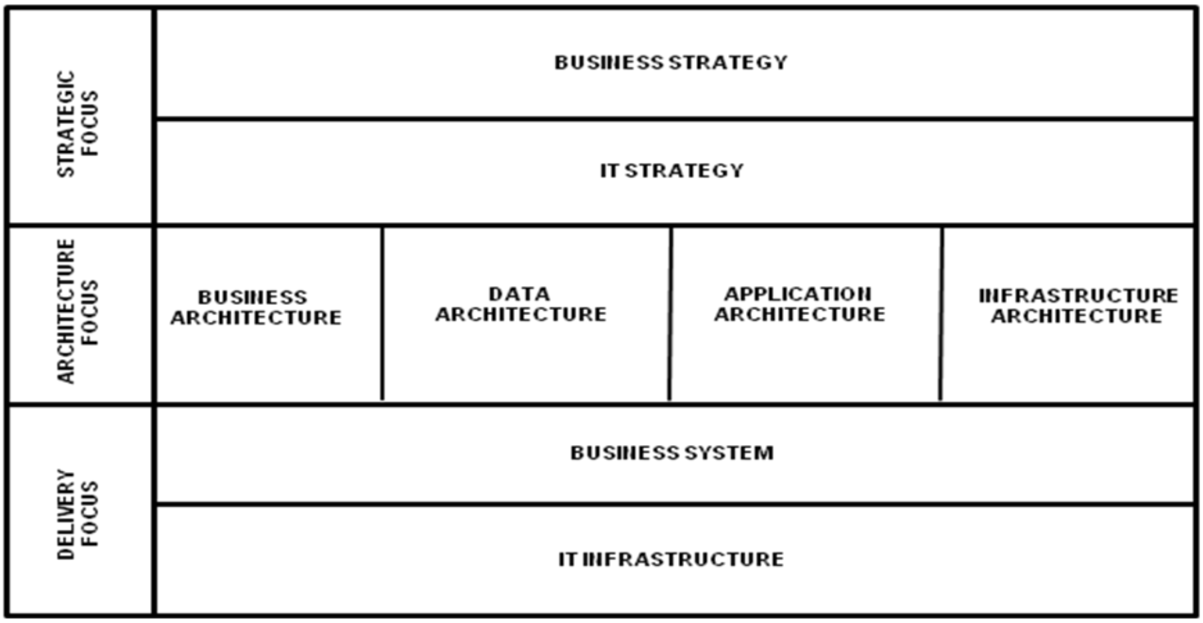


Figure 2.12. Situated Components of an Enterprise Architecture (Gerald, 2007)

In practice, Enterprise Architecture can usually be broken down into a number of component architectures. The following set of domain architectures are usually considered to be the fundamental components of any complete EA.

- A business architecture
- A data or information architecture
- An application architecture
- An infrastructure or technology architecture

Figure 2.12 shows how these components provide the ‘glue’ that connects an enterprise’s business and IT strategies to the delivery of its business systems and infrastructure

Some of the most popular enterprise architecture methods, frameworks and standards are listed in the Table 2.1. These are described in the following sections.

Table 2.1. Architectures, frameworks and standards

EA Methods/Processes	EA Frameworks	EA Standards
	Zachman Framework Section 2.6.2.1	FEAF Section 2.6.2.5
	CIMOSA Section 2.6.2.2	
	ARIS Section 2.6.2.3	
	TOGAF Section 2.6.2.4	

This review presents some of the most well known enterprise architecture approaches. However, an exhaustive review would be impossible. The observation made by the International Federation of Accountants FAC/IFIP Task Force on Architecture for Enterprise Integration in 1991, is probably just valid today. The observations are;

1. There are very large number of architectures or models already in the literature or developed as proprietary projects by many industrial groups
2. None of these were complete as yet
3. Most architectures present many of the same concepts but by means of different graphical and mathematical methods
4. The ancient parable of the group of blind Indian philosophers who attempted to describe an elephant after each had felt only different separate parts, certainly applies here – Each of the proposed architectures is describing the same subject but from widely varied and very incomplete viewpoints. Thus, the description appears to be very different. (William & Li, 1998)

Some EA methods, frameworks and standards have been deliberately excluded from this review because they do not appear to be very well developed or formalized. Other approaches have been excluded because they appeared to have been developed and presented in preliminary research papers, never to appear again. Consequently, it is likely that a good number of these approaches have found little acceptance or adoption outside of these environment.

The goal of this review however is to provide an understanding of the typical approaches to enterprise integration architecture and models within which, any understanding of the efficacy of an integrated information system must be situated. On this basis, the sample reviewed here has been selected in a way that identified trends and similarities between the various EA approaches.

2.7.2.1 The Zachman Enterprise Architecture Framework

Traditionally, systems have been developed in steps such as requirement analysis planning, system development and implementation. To substitute these methods Zachman (1987) published an enterprise architecture framework in 1987. In this framework, system development is based on different perspectives of various stakeholders. The Zachman framework is the most commonly used enterprise architecture framework (Evernden, 1996; Cook, 1996; Periasmy & Feeny, 1997; Morganwalp & Sage, 2003). The Zachman framework for enterprise architecture is a "common context for understanding a complex structure by identifying the generic structure that organizes, or classifies, the framework establishes a set of perspectives and common tools to define and describe an enterprise. The Zachman framework and related approaches identify the need to separate data, process and technology (Zachman, 1987; Finkelstein, 1992). It should be noted that Zachman does not distinguish between the terms data and information, but uses the term data in the same meaning as the information is uses in this study. However, the Zachman framework is strictly structure and does not include any elements to support the process of developing an enterprise architecture. This deficiency has faced some critics among researchers (Cook, 1996; Open Group, 2002; Ambler, 2002; Morganwalp & Sage, 2003).

The framework developed by John Zachman consists of a two-dimensional matrix. The columns represent various aspects of the enterprise that can be described or modeled; and the rows represent various viewpoints from which the aspects can be described. Thus, each cell formed by the intersection of column and a row represents an aspect of the enterprise modeled from a particular viewpoint. The six rows of the matrix include perspectives of (Hay, 2000):

i. Scope (Planner's view, contextual): Definition of the enterprise's direction and business purpose. This is necessary to establish the context for any system development effort.

ii. Enterprise model (Owner's view, Conceptual): This defines – in business terms – the nature of the business, including its structure, functions and organisation.

iii. System Model (designer's view, logical): This defines the business, but in more rigorous information terms. This perspective describes an application architecture.

iv. Technology model (Builder's view, physical): This describes how technology may be used to address the information processing needs identified in the previous rows.

v. Detailed representations (Sub-contractor's view, out of context): Here a particular language is chosen and the program listings, database specifications, networks etc are provided.

vi. Functioning system: A system is implemented and made part of an organisation.

The six columns address the interrogatives what (data aspect), how (function aspect), where (network aspect), who (people aspect), when (time aspect) and why (motivation aspect). The Zachman enterprise architecture framework is illustrated in Figure 2.13.

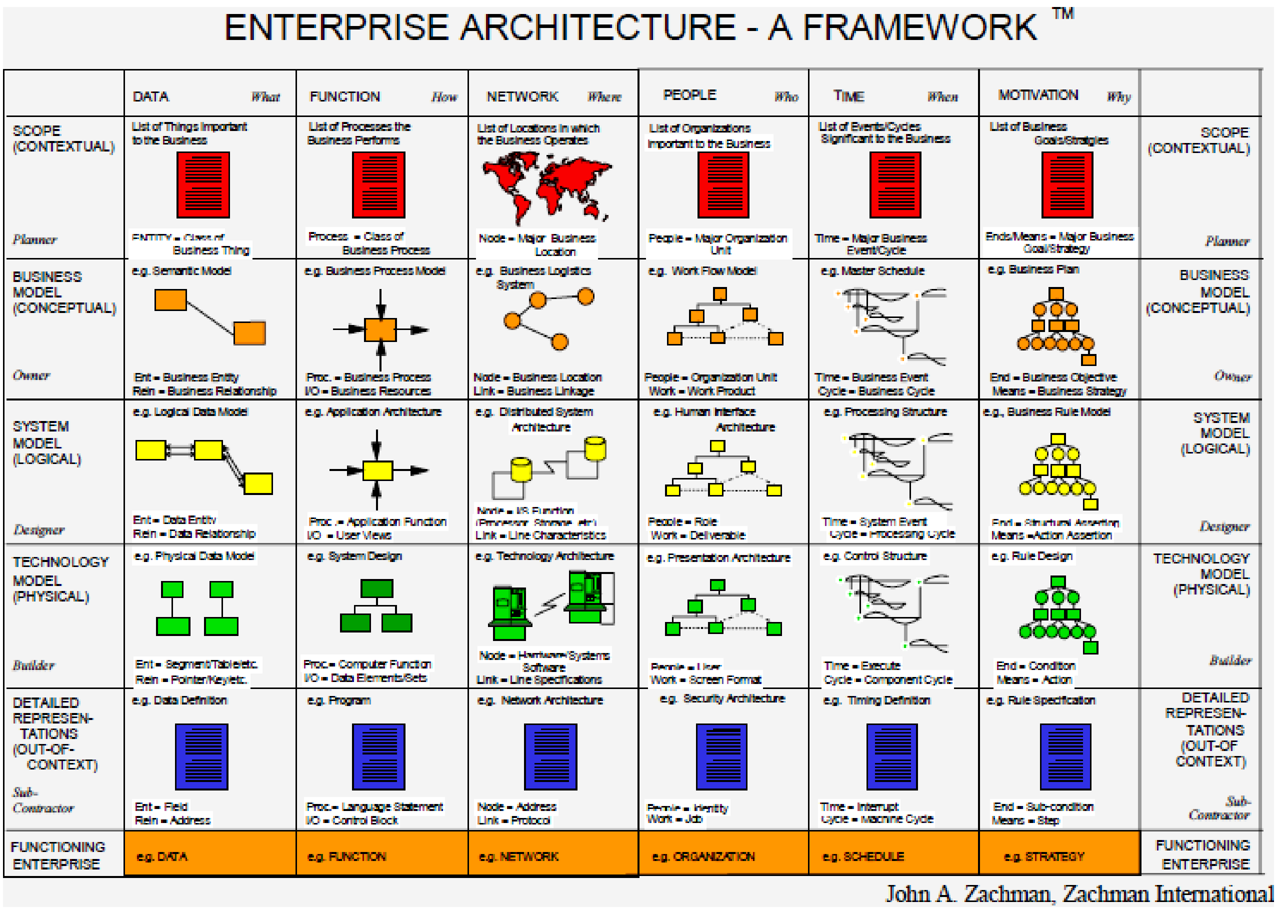


Figure 2.13. The Zachman enterprise architecture framework (Zachman, 1987).

Clearly, the strength of the framework is that it provides a natural way of thinking about an enterprise. The Zachman framework provides a comprehensive approach to enterprise architecture design with all the details within the enterprise. This level of details is an advantage of the framework but it is simultaneously a disadvantage because it makes the Zachman enterprise architecture framework quite complicated and heavy to use. The framework has the benefit of providing an overall picture of the enterprise context during the process of getting down to details. Another weakness of the framework is that it does not include any tools or guidelines to support the enterprise architecture building process itself. This is the most criticized weakness of the framework (Ambler, 2002; Vail, 2002; Morganwalp, 2003). This may cause sub-optimization and cost inefficiency if individuals involved in enterprise architecture development are working on their part of the architecture too independently. In addition to the lack of developing method, the Zachman framework does not acknowledge the possibility of using an application driven approach.

Although the Zachman framework has been widely accepted and used, there are some potential problems in it (Ambler, 2002; Morganwalp, 2003);

i. The Zachman framework can lead to a documentation-heavy approach. There are 36 cells in the framework and each one needs to be supported by one or more artifacts.
ii. The Zachman framework can lead to a process-heavy approach to development. To fulfill all the cells in the framework, the implementing process might be heavy.
iii. A major weakness of the framework is that it is strictly structural and it does not have any methods or reference models for development.

However, Ambler (2002) and de Villiers (2001) gave some guidelines on how to exploit the Zachman enterprise architecture framework efficiently:

i. Keep it simple
ii. Adopt the concept that your enterprise architecture efforts must reflect in a wide range of perspectives.
iii. Adopt the augmented form of the Zachman framework to avoid methodology bias.
iv. The Zachman framework is useful as a high level summary of relevant development artifacts and detailed descriptions of the relationships among artifacts, roles and activities.

2.7.2.2. CIMOSA (Computer Integrated Manufacturing Open Systems Architecture)

CIMOSA was developed for ESPRIT (European Strategic Program for Research and Development in Information Technology) by AMICE (a consortium of 30 major European vendors and users of CIM systems (e.g. IBM, HP, DEC, Siemens, Fiat, and Daimler-Benz). Original funding was provided by the European Common Market (EEC) through its ESPRIT project. CIMOSA defines a model-based enterprise engineering method which categorizes manufacturing operations into Generic and Specific (Partial and Particular) functions. These may then be combined to create a model which can be used for process simulation and analysis. The same model can also be used on line in the manufacturing enterprise for scheduling, dispatching, monitoring and providing process information.

Goals of CIMOSA: The two major goals of CIMOSA are;

(i) Help companies to manage change and integrate their facilities and operations

(ii) Help companies compete on price, quality and delivery time.

The CIMOSA architectural framework (Enterprise modeling and integration) includes:

(i) A general definition of the scope and nature of CIM

(ii) A description of the constituent systems and subsystems

(iii) A modular framework complying with international standards

An important achievement of CIMOSA is the idea of open system architecture for CIM made from vendor independent standardized CIM modules. Figure 2.14 shows the CIMOSA cube.

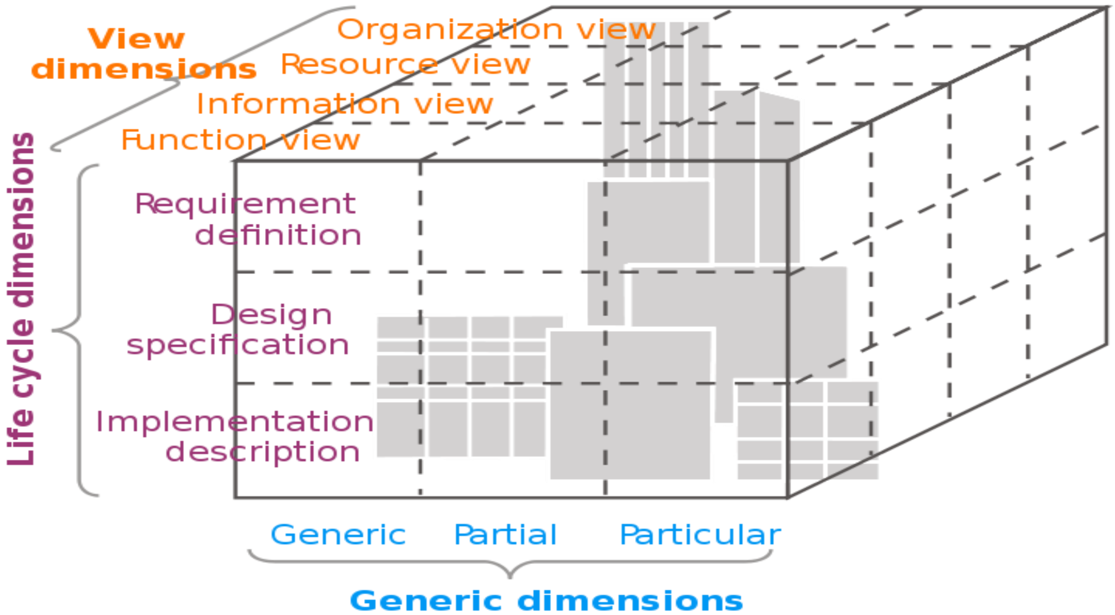


Figure 2.14 The CIMOSA cube (The Open Group, 2002)

CIMOSA defines four modelling views of the Enterprise Functions, namely; Function View, Information view, Resource view and Organisation view.

- The Function View describes work flows
- The Information View describes the Inputs and Outputs of Functions

- The Resource View describes the structure of resources (Humans, machines, and control and information systems)
- The Organization View defines authorities and responsibilities

CIMOSA also defines three modelling levels as;

(i) Requirement Definition – Standardized technical description of the company's concept.

(ii) Design Specification – Application of the technical requirement in the descriptive language.

(iii) Implementation – Description of the hardware and software components that will be used to apply the company concepts.

2.7.2.3. ARIS (Architecture for Integrated Information Systems)

Developed by Professor Scheer at the University of Saarbrucken in Germany, the overall structure of ARIS resembles CIMOSA, but focuses on the design of enterprise information systems. Therefore it provides specific modeling support for the information technology (IT) of the enterprise. ARIS supports enterprise modeling from an operational concept and Information Technology concept to Information Technology system implementation. It is structured in four views and three modeling levels.

The three modeling levels are the same with CIMOSA levels namely; requirements definition, design specification, implementation description.

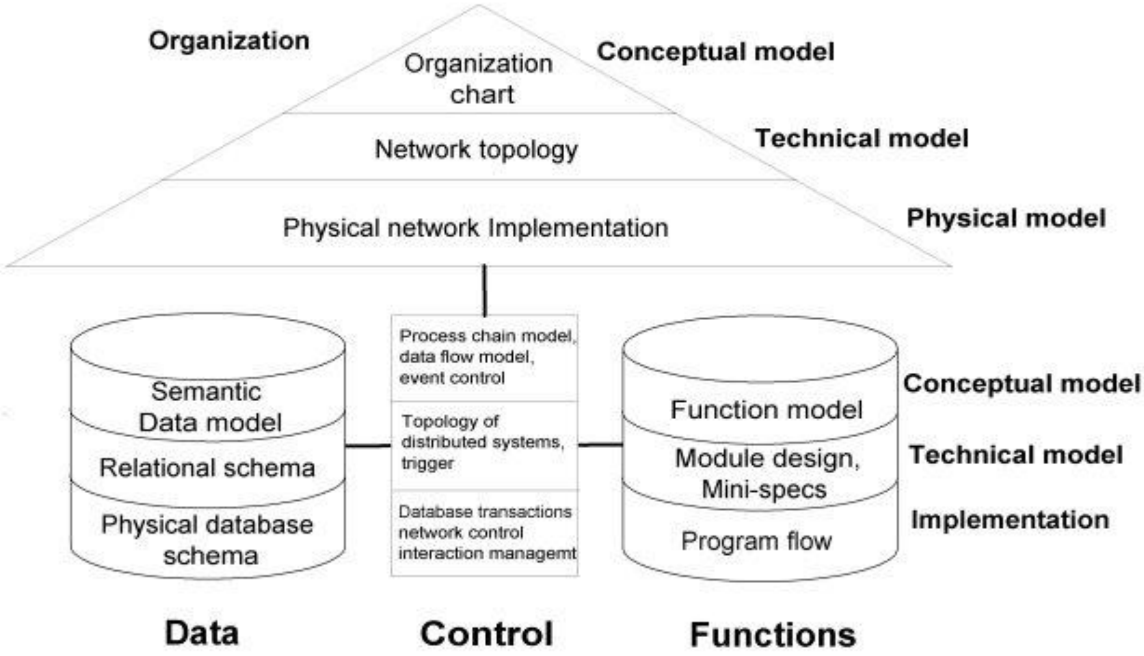


Figure 2.15. ARIS Architecture (Scheer, 1996)

The ARIS views are:

- **(i)** **Function View** – defines the model as a hierarchy of functions seen as program modules and generate the code for the programs.
- **(ii)** **Data View** – defines semantic data models (Entity-Relationship diagrams), translate them into relational schemata and implement them in physical databases.
- **(iii)** **Organization View** – defines enterprise structure, network topology and physical implementation.
- **(iv)** **Control View** – federates the architecture; here business processes are put together and implement as logical sequence of program.

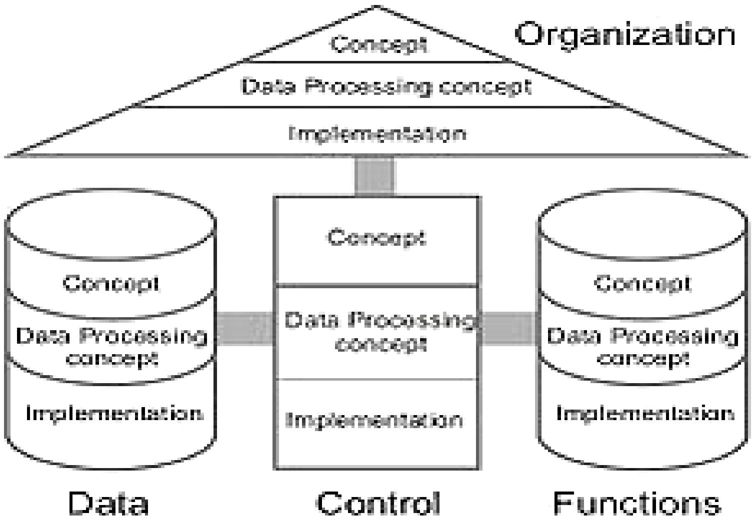


Figure 2.16 Model of the ARIS Framework (Scheer, 1996)

2.7.2.4 TOGAF (The Open Group Architecture Framework)

TOGAF is a framework for developing an enterprise architecture. It is described as a set of documentation published by The Open Group (The Open Group, 2002). The original development of TOGAF Version 1 in 1995 was based on the Technical Architecture Framework for Information Management (TAFIM), developed by the US Department of Defense (The Open Group 2002). "The Open Group is an international vendor and technology neutral consortium that is committed to delivering greater business efficiency by bringing buyers and suppliers of information technology to lower the time, cost and risk associated with integrating new technology across the enterprise" (The Open Group, 2002). TOGAF is not an architecture in and of itself, rather it provides the necessary tools with which an organisation can produce an architecture for itself (Beveridge & Perks, 2000). This is because TOGAF is an enterprise architecture framework and not an enterprise architecture. The major difference between Zachman framework and TOGAF is that TOGAF includes guidelines and tools on how to build an enterprise architecture. In TOGAF, this is called the Architecture Development Method (ADM) and it is an essential part of the framework. The Zachman framework does not include a development method.

As the Zachman enterprise architecture framework, TOGAF also supports the Concepts of stakeholder views. However, in TOGAF, different stakeholder views do not have as important a role as in the Zachman framework. According to TOGAF (The Open Group, 2002), the following stakeholder views should be considered.

- Users
- Systems and Software Engineers
- Operators, Administrators and Managers
- Acquirers

The architecture views to correspond with the above stakeholder views include (The Open Group, 2002):

i. Business Architecture Views, which address the concerns of the users of the system and describe the flows of business information between people and business process.
ii. Data Architecture Views, which address the concerns of database administrators and system engineers responsible for developing and integrating the various database components of the system.
iii. Application Architecture Views, which address the concerns of system and software engineers responsible for developing and integrating the various application software components of the system.
iv. Technology Architecture Views, which address the concerns of acquirers (procurement personnel responsible for acquiring the commercial-off-the-shelf (COTS) software and hardware to be included in the system), operations staff, systems administrators and systems managers.

It should be noted that as well as the Zachman framework, TOGAF does not distinguish data and information. TOGAF is designed to support four types of architecture. These are commonly accepted as subsets of an overall enterprise architecture (The Open Group, 2002):

- A business architecture – this defines the business strategy, governance, organisation and key business processes.

- An application architecture – this kind of architecture provides a blueprint for the individual application. Systems to be deployed, their interactions and their relationships to the core business processes of the organisation.
- A data architecture – this describes the structure of an organisation's logical and physical data assets and data management resources.
- A technology architecture – this describes the software infrastructure intended to support the deployment of core, the mission-critical applications. This type of software is sometimes referred to as "middleware".

The Open Group updates TOGAF quite frequently and the current release of TOGAF includes three main components (Morganwalp, 2003): the TOGAF Standard Information Base (SIB), the TOGAF Technical Reference Model (TRM) and the TOGAF Architecture Development Method (ADM). Together SIB and TRM form an Enterprise Continuum, which is a "virtual repository" of all the architecture assets – models, patterns and architecture descriptions – that exist both within the enterprise and in the IT industry at large, and which the enterprise considers to have available for the development of architectures (TOGAF, 2002). In addition to SIB and TRM, the Enterprise Continuum includes the Integrated Information Infrastructure Reference Model (IIIRM), which is specifically aimed at helping the design of architectures that enable and support the information flow without boundaries.

The Standards Information Base (SIB) is a database of industry standards that can be used to define the particular services and other components of an enterprise –specific architecture. TOGAF Technical reference Model (TRM) provides a model and taxonomy of generic platform services. The TRM also identifies system-wide capabilities such as security and network management (Morganwalp, 2003).

TOGAF's Architecture Development Method (ADM) is a cyclic and iterative method to support the enterprise architecture building process. ADM as shown in Figure 2.17 consists of eight phases.

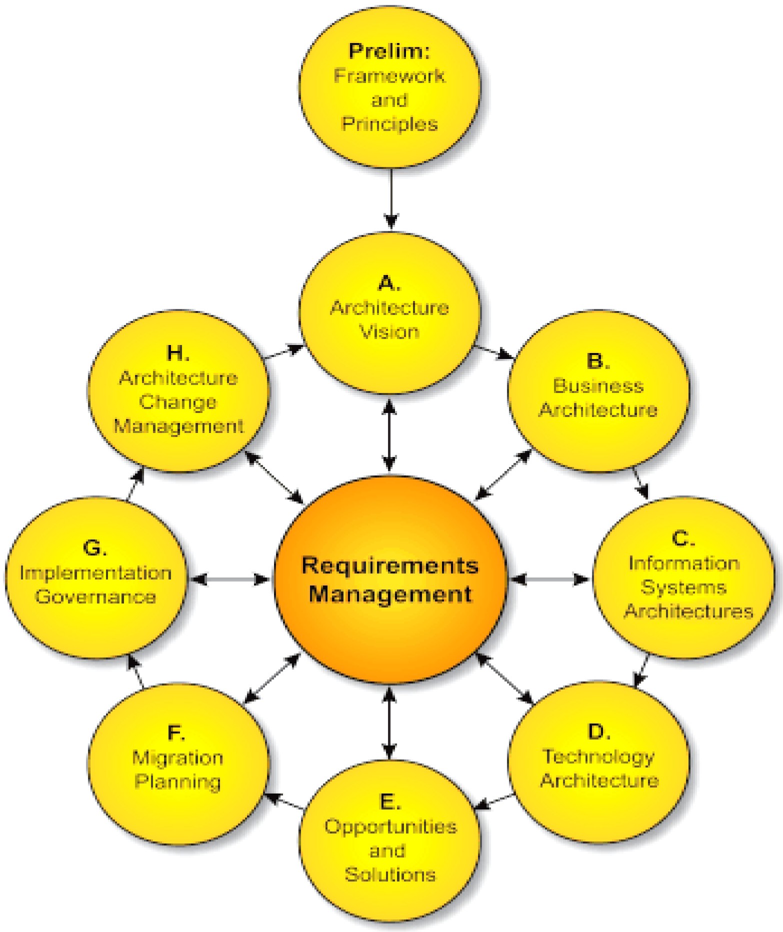


Figure 2.17. TOGAF Architecture Development Method (The Open Group, 2002).

At relevant places throughout the TOGAF ADM, there are reminders to consider which, if any, architectural assets from the Enterprise Continuum the architect should use (The Open Group, 2002). Characteristics for TOGAF is that architectures are driven top down from a business stand point, not bottom-up from a technical point of view (Beveridge & Perks, 2000). TOGAF Architecture development Method (ADM) describes a step-by-step approach to develop an enterprise architecture. ADM is a generic model and each organisation should decide if they want to customize it for specific needs or exploit the generic model (The Open Group, 2002). Developing an enterprise architectures with the ADM is an interactive process, which consist of eight stages. The stages of ADM are illustrated in Figure 2.17. In the first phase the strategic business requirements are defined and relevant stakeholders identified. Also an architecture vision to meet these requirements is determined.

In the second phase, the current baseline business architecture is described and target business architecture developed. This phase is used to analyze the gaps between the baseline and the target business architectures. The objective of the third phase is to develop target architectures covering the Data and Application system domains (The Open group, 2002). Depending on the scope of the project, data and application architectures can be created as sub-phases. Spewak (2000) recommends a data-driven approach for the enterprise architecture building process. Regarding the data, the target is to define the major types and sources of data necessary to support the business, but this does not mean database design or the design of any logical or physical storage systems. Concerning the applications, the objective is to define what major applications are required to process the data and support the business. However, this excludes applications system design itself and concentrates on the kinds of applications that are relevant to the enterprise (The Open Group, 2002).

The fourth phase in ADM is technology architecture development. This phase consists of eight sub-phases: create baseline, consider different views, create architectural model, select services portfolio, confirm business objectives are met by architectural model, determine criteria for specifications, define architecture completely and conduct gap-analysis (Morganwalp, 2003). The fifth phase, opportunities and solutions, identifies the strategic change and the top-level projects to be undertaken in moving from the current environment to the target architecture (The Open Group, 2002). The Migration planning phase is for developing the various project plans for projects, which have to be implemented. This phase is also for prioritizing these plans. The objective of the seventh phase, implementation governance, is to formulate recommendations and plans for each implementation project and then implement and deploy the systems (Morganwalp, 2003).

The last phase of ADM, Architecture Change Management, includes creating a maintenance procedure for the new baseline that has been implemented in the previous phase.

2.7.2.5 FEAF (Federal Enterprise Architecture Frameworks)

The Federal Enterprise Architecture Framework (FEAF) was developed by the council of Chief Information Officers of the US Government in the 1990’s. FEAF provides direction and guidance to federal agencies for structuring an enterprise architecture. The FEAF is a strategic information asset base that defines the business, information necessary to operate the business, technologies necessary to support the business operations and transitional processes for implementing new technologies in response to the changing needs of the business. The target of FEAF is to promote shared development for common federal processes, interoperability and sharing of information among federal agencies and other governmental entities. The focus of the federal enterprise architecture is limited to common federal architecture issues, which benefit federal organizations and the public, if resolved at the federal level (CIO Council, 1999).

FEAF combines structure with the process overview in a transition process from the baseline enterprise architecture to the target enterprise architecture. FEAF is organized in eight components. However, it does not provide detailed steps on how to accomplish those components. Therefore, it differs from TOGAF, which gives detailed guidance for the enterprise architecture development process. FEAF has four levels and each of them is progressively more detailed than the previous level. In FEAF, the eight components of an enterprise architecture are (CIO Council, 1999):

- Architecture Drivers – Represent two types of external stimuli or change agents for the enterprise architecture: business and design. The business drivers could be new legislation, new administration initiatives, budget enhancements for accelerated focus areas and market forces. Design drivers include new and enhanced software and hardware and their combinations with a variety of deployment approaches.
- Strategic Direction – Guides the development of the target architecture and consists of a vision, principles, goals and objectives.
- Current Architecture – Defines the "as is" enterprise architecture and consists of two parts: current business and design architectures (i.e. data, applications and technology). This represents the current capabilities and technologies and is expanded as additional segments are defined.
- Target Architecture – Defines the "to-be-built" enterprise architecture and consists of two parts: target business and design architectures (ie. Data, applications and technology). This represents the future capabilities and technologies resulting from design enhancement to support changing business needs.
- Transitional Processes – Support the migration from the current to the target architecture. Critical transition processes for the federal enterprise include capital IT

investments planning, migration planning, configuration management and engineering change control.

- Architectural Segments – Consists of focused architecture efforts on major cross-cutting business areas, such as common administrative systems; program areas, such as trade and grants; or small purchases via electronic commerce. They represent a portion (segment) of the overall enterprise architecture. A segment is considered to be an enterprise within the total federal enterprise.
- Architectural Models – Define the business and design models that comprise the segments of the enterprise description.
- Standards – Refer to all standards (some of which may be mandatory), guidelines and best practices.

In Figure 2.18, relationships between these components are shown (CIO Council, 1999). Figure 2.18 describes level three of FEAF. As shown in Figure 2.18 the FEAF building process is divided in two parts; business drivers and design drivers.

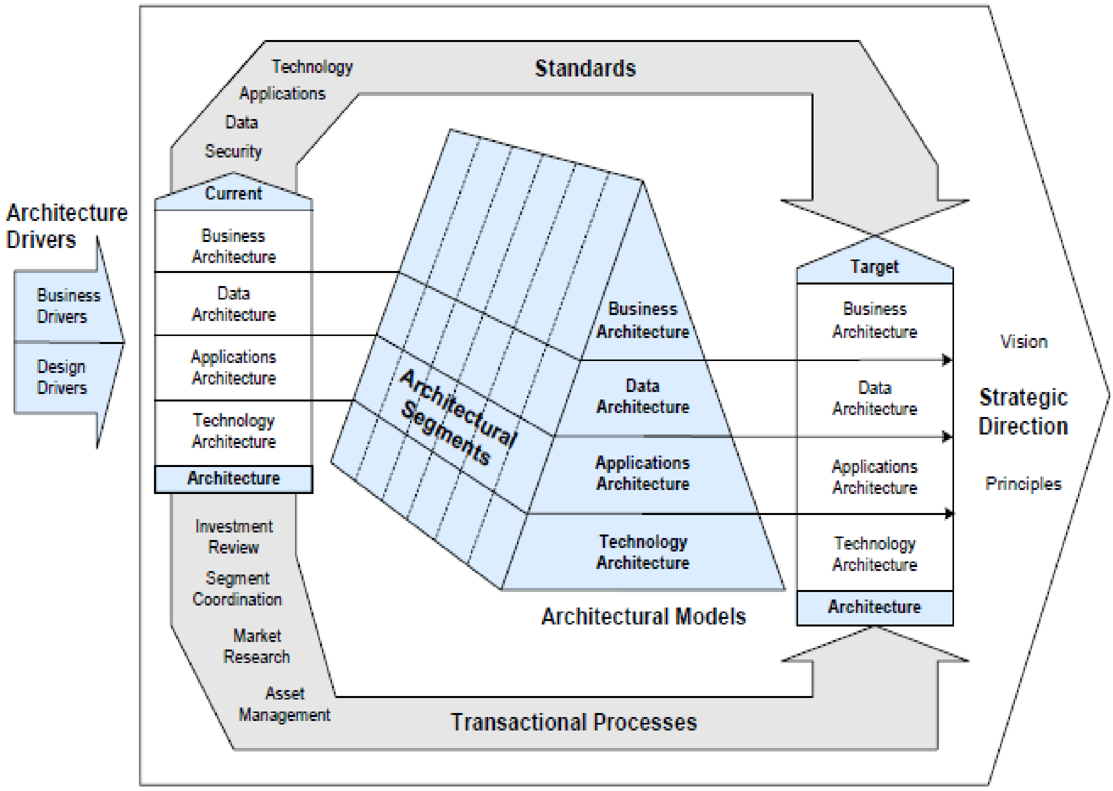


Figure 2.18 Federal Enterprise Architecture Framework (Level 3) (CIO Council, 1999)

As illustrated in figure 2.18, the FEAF partitions a given architecture into business, data, applications and technology architecture. Enterprise architecture work starts from a

current architecture and as a result of enterprise architecture development work is a target enterprise architecture. The FEAF does not distinguish the terms data and information.
At level four, the FEAF is quite similar to the Zachman framework. FEAF level four describes the stakeholder views (in rows) and interrogatives what, how and why as columns. The FEAF includes the first three columns of the Zachman framework. FEAF level four is illustrated in Figure 2.19.

	Data architecture (entities=what)	*Application Architecture (activities=how)*	**Technology Architecture (locations=where)**
Planner Perspective	*List of Business Objects*	*List of Business Processes*	List of Business Locations
Owner Perspective	*Semantic Model*	*Business Process Model*	Business Logistics System
Designer Perspective	*Logical Data Model*	*Application Architecture*	System Geographic Deployment Architecture
Builder Perspective	Physical Data Model	Systems Design	Technology Architecture
Subcontractor Perspective	Data Dictionary	Programs	Network Architecture

Figure 2.19 FEAF Architecture Matrix (CIO Council, 1999).

Cells which are in the focus of this study are highlighted. The cells that this study primarily covers are highlighted in Figure 2.19.

2.8 Bringing Enterprise Architecture to SMEs

Although many new firms start up each year, survival becomes increasingly difficult (Levy & Powell, 2005). Because of the many challenges lying ahead of them, only 70 percent of SMEs survive at least 2 years, 50 percent at least 5 years, a third at least 10 years, and only a quarter stay in business 15 years or more (Bureau of Labor Statistics US, 2011, Bernaert et al., 2013)
According to Hambrick and Mason (1984), company survival success is conditional on the knowledge of the entrepreneurs. Literature agrees on this importance of knowledge management as one of the sources of sustainable competitive advantages (Lybaert, 1998). In small and medium-sized enterprises, the entrepreneurs mostly present themselves in the form of one person, combining both the ownership and managerial function (Omerzel & Antoncic, 2008). It is this person's responsibility to steer the company in the desired

direction by adapting processes to pursue this direction, and to provide clear communications with employees so that the entire company is focused on reaching the desired direction.

Some of the problems that can explain the high SME failure rate can be related to a weakness of the entrepreneur's knowledge management, causing a lack of structure and overview of the company (Proper and Greefhorst, 2011). Bernaert et al. (2013) lists a number of problems that can occur caused by such a lack of overview:

• For ERP adoption, the most important criterion used by SMEs in selecting an information system is the best fit with current business procedures (van Everdingen et al., 2000). If the CEO leaves the company for some reason (e.g., he sells the company or a child takes over), his overview of the company has to be transferred to the new CEO.

• In an SME, employees tend to know less about the structure of the company and why things are done. Although the entrepreneur knows the overview of the company, it is difficult for him to communicate with its employees about strategic issues without having an explicit overview (Kamsties et al., 1998).

• A concrete job description and overview of tasks and responsibilities of employees is difficult to keep track of, especially in a changing environment and enterprise (Kamsties et al., 1998).

• A strategy is not static, neither are processes. Keeping processes at all time in line with the strategy is difficult to achieve (Dougherty, 1992).

• An SME has different stakeholders with different desires and goals. Balancing these goals as good as possible is not a simple assignment. (Heyse et al. 2012).

2.8.1 Requirements for EA for SMEs

To guide the development and evaluation of an EA approach for SMEs, requirements for an appropriate solution are needed. First, the requirements for EA in general are presented, followed by those for the adoption and successful use of IT in SMEs. To end, the combination of these two sets of requirements into a single set for EA in an SME context is also described.

2.8.1.1 Requirements for Enterprise Architecture (EA)

The essential requirements for EA (Lankhorst 2013; Zachman 1987) are the following:

1. **Control**: EA should be usable as an instrument in controlling the complexity of the enterprise and its processes and systems.
2. **Holistic Overview:** EA should provide a holistic overview of the enterprise and be able to capture its essence: the stable elements that do not vary across specific solutions found for the problems currently at hand.
3. **Objectives:** EA should facilitate the translation from corporate strategy into daily operations.
4. **Suitability:** EA should be suitable for its target audience. It needs to be understood by all those involved, even if they come from different domains.
5. **Enterprise-wide:** EA should enable optimization of the company as a whole instead of doing local optimization within individual domains.
6. **Integrated**: EA should enable joining all partial architectures together into a whole (By Author).

The fourth requirement refers to the target audience. In our case, the target audience is SMEs and, more specifically, their owners or managers. Therefore, requirement 4 is refined using the requirements for the adoption and successful use of IT in SMEs

2.8.1.2 Requirements for the Adoption and Successful Use of IT in SMEs

The requirements for the adoption and successful use of IT in SMEs with respect to number 4 in (2.8.1.1) are as follows:

4.1 The approach should enable SMEs to work in a time-efficient manner on strategic issues.

4.2 A person with limited IT skills should be able to apply it.

4.3 It should be possible to apply the approach with little assistance of external experts.

4.4 The approach should enable making descriptions of the processes in the company.

4.5 The CEO must be involved.

4.6 The expected revenues of the approach must exceed the expected costs and risks.

By combining these requirements with the EA requirements, this study obtained a set of requirements for the adoption and successful use of EA in SMEs.

According to requirement 4 and thus 1 - 6, the EA model should be understandable and adaptable by non-EA experts in SMEs.

EA has become one of the top priorities of IT executives and is considered an important instrument for aligning the required changes in corporate strategy and business processes with an increasingly complex IT landscape (Luftman & Ben-Zvi 2011). Some of the most recognized benefits of EA are that IT can be used more efficiently and flexibly, business and IT can be better aligned (Radeke 2011; Tamm et al. 2011), and a better fit between business operations and strategy can be achieved (Hoogervorst 2004). Braun and Winter (2005) emphasized that in order for business-IT and strategy to be aligned, EA must be adaptable and constantly held up-to-date.

SMEs constitute over 90 % of operating businesses in many countries, in the U.S. even 99.7 % (Small Business Administration, 2011) , in Europe 99.8 % (European Commission, 2011) and in Nigeria about 90% (Mordi, 2005). There is therefore a great need for more rigorous research that is relevant for this important sector of the economy (Devos 2011).

Right now, existing EA frameworks are primarily used in large enterprises (Gartner 2012). Wißotzki and Sonnenberger (2012), among others, recognize the importance of EA and EA management (EAM) in particular, but also notice that EAM is still mostly unexplored and rarely used, especially in the context of SMEs (Devos 2011). Yet, such specific research is crucial, as research findings based on large businesses cannot be generalized to small businesses due to the inherent differences between SMEs and large businesses (Aarabi et al. 2011).

Lybaert (1998) discovered that SME owners or managers with a greater strategic awareness use more information and that SMEs that use more information are generally more successful. Hannon and Atherton (1998) further revealed that for SMEs success is correlated with higher levels of strategic awareness and better planning of owners-managers. In addition, there is evidence to believe that companies that make strategic rather than just financial business plans perform significantly better financially than those that do not (Smith, 1998).

Jacobs et al. (2011) argued that from the perspective of change and complexity, EA could assist SME management during the growth of a small enterprise. For example, according to Aarabi et al. (2011), ERP (Enterprise Resource Planning) systems cannot be successfully implemented and utilized in SMEs if EA is disregarded. In fact, it is EA's integration of strategic goals, business processes, and technology planning methods that provides the standards, roadmaps, and context for ERP implementation (Zach, 2012). As Bidan et al. (2012) concluded, process standardization in SMEs is more important than the deployment of technology (e.g., ERP systems) to improve organizational performance. In short, SMEs need to get a structured view of their company, even before they start implementing an ERP solution.

Hence, while EA might offer SMEs a solution to typical problems related to a lack of overview, strategic awareness, IT planning, and business-IT alignment, EA approaches that cater for the specificities of small businesses are still missing. This lack of research on an EA approach that can readily be used for SMEs is exactly the problem that is addressed in the present research.

2.9 Information Requirements and Information Architecture

Identifying information requirements is one of the key phases in planning and implementing an information system. Equally, it is important in planning the whole architectures of enterprise systems, in this case business application architecture. Browne and Rogich (2001) defined requirements determination as the process of gathering and modeling information about required functionality of a proposed system by a systems analyst. Information requirement should be aligned with business requirements in the enterprise. However, a number of authors have recognized that correct and complete information requirements are frequently difficult to obtain (Leifer et al., 1994). Various methods, tools, and techniques have been proposed to improve the information requirements determination process (Regnell et al., 2001). Information modeling is closely related to the information requirements determination process. Information modeling is the process of formally documenting the problem domain for the purpose of understanding and communication among stakeholders (Siau, 1999). Information models, which are mostly graphic, are used to represent both static phenomena (things and their properties) and dynamic phenomena (events and processes) in some domain (Wand & Weber, 2002). According to Kung and Solvberg (1986), information models have at least four purposes:

1. Supporting communication between developers and users
2. Helping analysts understand a domain
3. Providing input for the design process
4. Documenting the original requirements for future reference

Despite these efforts the information requirement determination process remains a challenging task. Davis (1982) suggested three main reasons for the difficulty of obtaining a correct and complete set of requirements:

i. The constraints on human as information processors and problem solvers
ii. The variety and complexity of information requirements
iii. The complex patterns of interactions among users and analysts in defining requirements.

Consequently, if systems are developed based on information requirements, which are either wrong or incomplete, the systems themselves are likely to be unsuccessful. A complete and accurate collection of user requirements sets the stage for an efficient development process that increases the likelihood of a successful organizational system (Davis, 1982). In addition to issues listed by Davis, the constant change of business and business environments hamper the obtaining of information requirements. Although Davis aims at a complete set of requirements, in practice it seems impossible to achieve and there are always factors of uncertainty to be considered.

One outcome of the information requirements defining process is an information architecture (some authors used the term data architecture instead). According to Brancheau et al. (1989) "An information architecture is a high-level map of the information requirements of an organisation. It shows how major classes of information are related to major functions of the organisation. In its pure form, information mapping is independent of personal staffing, organisation structures and technology platforms". Niederman et al. (1991) made a study among IT executives concerning the top 10 priorities for them at the time. In this survey, information architecture was ranked as number one. Although this survey was made more than ten years ago, it seems clear that information architecture would still be on the top list. Information architecture can be used to guide applications development and to facilitate the integration and sharing of information. According to Niederman et al. (1991) several authors agreed that information architecture offers the potential to serve as a basis for building responsive and long lasting set of business applications, in other words a business application architecture (Scheer, 1989; Martin, 1989). However, Niederman et al. (1991) continued that information architecture is difficult to capture, use and maintain, due to both breath of information requirements and the changing nature of the business environment. According to Van Den Hoven (2003), most enterprises do not have a well-integrated view of their data. As a result, data is fragmented, inconsistent, redundant and generally inaccessible beyond the business function that it is directly supporting. Umar et al. (1999) stressed the importance of ensuring that quality of data is part of the information architecture designing process.

According to Van Den Hoven (2003), an effective information architecture enables:

i. Data to be used more efficiently and effectively within the enterprise
ii. Better integration of data to get an overall understanding of the enterprise and how well it is performing.
iii. Better integration of the enterprise with its customers, suppliers and business partners.

An information architecture establishes a base for the integrated business application (enterprise system) architecture and basic rules for how and where and which data entities should be maintained. Well-designed information architecture provides possibilities to build a high-quality business application architecture.

Gosling (1993) listed some characteristics of a good information architecture:

i. Inclusiveness: A good information architecture should be as comprehensive as possible, including as much of the information required to carry out its function as possible.
ii. Balance: An information architecture should provide a good balance between data consistency and flexibility. In other words, it should enforce some uniformity

in the classification of information while allowing flexibility to add new information that might be specific to one part of the organisation. It must also be flexible enough to accommodate new information that will certainly arise.

iii. Multiple views: The user should be able to look at the information in a variety of ways.

iv. Business orientation: An information architecture should reflect the organizational structure and orientation of the company, accommodating a line of business focus as well as traditional hierarchical relationships.

2.10 Information System Investments

Applications classification models and enterprise architecture development methods are linked inseparably to information system investments because when developing an enterprise or application architecture, investments are always required. There have been a number of studies on how IT investments should be assessed and justified; there is no common understanding how it should be done (Davern & Kaufman, 2000). Both the costs and benefits of information system investments are difficult to measure because they are often intangible in nature and the benefits are realized during a long period of time (Lederer &Mendelow, 1993). Other analysts suggested that traditional financial analysis techniques are commonly in use, but that companies are finding it increasingly difficult to use them because the types of benefit become more difficult to quantify adequately (Ballantine et al., 1994). Also, the benefits from information system investments may be seen differently by different stakeholders or persons involved (Farbey et al., 1992). In addition, sometimes it is difficult to verify the causality between information system investment and benefits.

Parker and Benson (1992) assessed in detail the ways in which information and systems benefits accrue and how they can be quantified to help in justifying investments. They considered three main types of application investments:

i. Substitutive – technology replacing people with economics being the main driving force, improves efficiency.

ii. Complementary – improving organizational productivity and employee effectiveness by enabling work to be performed in new ways.

iii. Innovative – achieving competitive edge by changing trading practice, creating new markets etc.

They continued by suggesting five different techniques for evaluating information systems investment benefits:

1. Traditional Cost-benefit analysis focuses on efficiency improvements in company processes resulting in automation.
2. Value linking estimates the improvement in business performance, not just in the savings made by improving linkages between processes or activities (eg automatic reconciliation of orders).
3. Value acceleration considers time dependencies of benefits and costs in other departments of system improvements (e.g. giving sales data to buyers on daily bases and improving their ability to respond to changes in demand).
4. Value restructuring considers the productivity resulting from process and organizational change and change of job roles.
5. Innovation evaluation attempts to estimate the value of new business or new business practices levered from IT/IS (eg. Launch of on-line banking services).

Mooney et al. (1995) proposed a business process oriented approach in the evaluation of the potential value of IT in business. They divided business processes into operational and managerial processes. In their framework, IT business value can be automational, informational or transformational. Automational refers to efficiency of processes and informational means better sharing or better quality of information. Transformational refers to the value created by facilitating process innovation. The framework by Mooney et al. (1995) is illustrated in Figure 2.20.

	Dimensions of IT Business Value		
Business Processes	Automational	Informational	Transformational
Operational			
Management			

Figure 2.20. A framework for Identifying business value created through the impact of IT on business processes dimensions of IT Business Value (Mooney et al., 1995)

Equally important as the evaluation of information system investments as such, is the capability to prioritize them. This is because it is unlikely possible to execute all application investments in parallel. Ward and Peppard (2002) suggested three factors to be included when assessing the priorities for information system investments. First, what is the most important to do, based on identified benefits. Second, what can be done based on resources available and third, what is likely a success based on the risks of failure of each investment. According to the research of Farrel (2003), three practices distinguish the companies that were most successful in their IT investments. First, they targeted their investments at the productivity levels that mattered most to them and their industries. Second, they carefully thought through the timing and sequence of their investments. Third, they didn't pursue IT in isolation but rather developed managerial innovations in parallel with technology innovations.

Prioritizing information system investments depends also on the company's overall value position and choice of pursuing competitive advantage (e.g. cost leadership, customer intimacy or product leadership). Ward and Peppard (2002) suggested applying McFarlan's (1984) application classification model logic also in this respect:

i. Strategic applications should relate to the dimension in which the company seeks to excel in one to three years (e.g. operational excellence or product leadership), with the objective of gaining advantage in the market place.
ii. Key operational application investments are essential in any dimension if the systems are causing performance level problems.
iii. Investments on high-potential applications would normally be prototypes related to specific strategic developments or evaluation of ideas relevant to other dimensions.

However, even if some good prioritizing techniques would exist, Lederer and Mendelow (1993) argued that the reasons for establishing and setting priorities for information system investments can be divided into two groups: rational and political. Clemons (1991) emphasized that the increasing competitive impact of information systems makes evaluation and prioritizing even more problematic. It is difficult to evaluate the value of building strategic information systems because the environment of contemporary companies is characterized by turbulence and constant change.

As a part of the appraisal of information system investment viability, it is critical to assess the potential risk. The risk may be realized as failing to deliver anything at all, failing to deliver the expected benefits or exceeding the schedule or budget significantly. Study on information system investments by Lyytinen and Hisrchheim (1987) suggested that failure can occur in four domains:

1. Technical failure – this is a domain of the IT department that is responsible for the technical quality of the system and the technology it uses.
2. Data failure – this is a shared responsibility between the IT department and the users who input the data.
3. User failure – this may be caused by the users who cannot use the system or partly caused by information system professionals, because insufficient training or documentation. A major weakness in many projects is inadequate training.
4. Organizational failure – systems may be satisfactory for particular functional needs but fail to satisfy the organisation overall. This happens, because of inadequate understanding of how the system relates to other processes and activities.

Although literature concerning IT investments deals mainly with information system investment and not specifically business application investments, all that has been suggested applies also to business application investments. This is because business applications are defined as part of information systems. It can even be argued that business applications are the most complex part of information systems and thus the management of business application investments is the most difficult part of information system investments. Therefore, the difficulties to assess information system investments apply also to business application investment. Also, it can be argued that the most significant benefits of information system investments can be achieved with business application investments because they have the closest link to the business. Thus, in terms of potential benefits, business applications investments are probably the most interesting part of the system investments.

2.11. Enterprise Application Integration

Organisations use a diversity of information systems such as ERP, CRM and e-business applications to support their business processes. This diversity of heterogeneous and in many cases incompatible systems are causing problems in terms of integrating applications. Integrating business applications efficiently is a vital part of building a business application architecture. Typically business processes go through several applications and without integration the possibility to automate those business processes is limited. Enterprise Application Integration (EAI) is attempting to overcome such challenges. Linthicum (1999) defined enterprise application integration as: "Unrestricted sharing of information between two or more enterprise applications. A set of technologies that allow the movement and exchange of information between different applications and business processes within and between organisations".

Applications integration is achieved through the incorporation of functionality from disparate systems using a variety of integration technologies such as adapters and message brokers (Themistocleous & Irani, 2002). According to Inmon (1999) there was a clear demand for EAI particularly as ERP applications became more prevalent in the 1990's and there was a need to be able to integrate already existing applications and data within the ERP system. Although legacy systems have problems with maintainability and inflexibility, they provide reliable solutions and therefore the amount of legacy systems in use remains high (Lloyd et al., 1999). This implies that the importance of enterprise application integration between legacy systems and recently implemented applications will continue. Themistocleous and Irani (2001) reported that a significant business benefit of application integration is the reduction of overall integration cost, because it provides a flexible, manageable and maintainable enterprise infrastructure that supports changing business and technical requirements. According to Ruh et al., (2000) based on an integrated enterprise architecture, companies can increase their relationship with their clients. Also Gable (2002) emphasized the cost savings attained with enterprise application integration. The savings come through lower cost per transaction, decreased error correction cost and decreased need to write and maintain customized interfaces between individual applications.

In building effective business application architectures, enterprise application integration has a valuable role to ensure seamless integration of various applications in the architecture. If an enterprise application integration really works as it is planned, it allows flexibility in choosing applications for the business application architecture because integration can be achieved without using the same enterprise wide application, such as all inclusive ERP, for all functions.

2.11.1. Strategies for Enterprise Application Integration

It is noted that enterprise applications are not implemented on one operating system using one computing language. The individualism of enterprise applications has to be recognized. Under this scenario, software integration can take the following strategies (Linthicum, 2001) and these strategies are shown in Figure 2.21.

Basics

- ***Data oriented:*** This focuses on the equivalence mapping between two databases, both in syntax and semantics. This approach does not change the application code that makes use of data in databases.
- ***Application-interface oriented:*** This refers to the leveraging of the interfaces exposed by applications. Developers leverage these interfaces to access both business processes and simple information. In order to integrate the applications, the interfaces must be used to access both process and data, extract the information, place it in a format understandable by the target application, and transmit the information. Message broker is the perfect solution for this type of software integration strategy.
- ***Method oriented:*** This is the sharing of the business logic that exists within an enterprise. The applications may access each other's methods without having to rewrite each method within the respective application. The solutions include distributed objects, application servers, and transaction processing monitors.
- ***Portal oriented:*** This approach is used by application architects to integrate applications by presenting information from several local or partner applications within the same user interface.

- ***Process integration oriented:*** This provides an abstract business-oriented layer on the top of more traditional information movement mechanisms. This also provides process automation features, e.g., a view of how business information flows between different applications. This does not typically deal with physical integration flows and physical systems, but with abstract and shared processes.

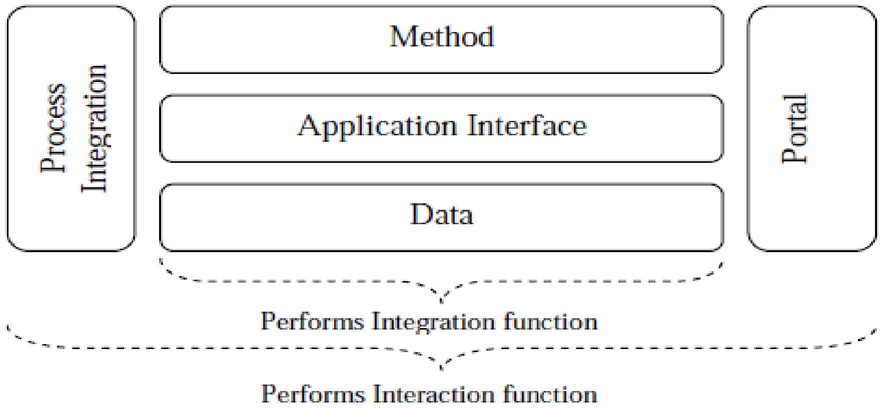


Figure 2.21. Application integration approaches (Linthicum, 2001)

2.11.2 Middleware

Middleware is a simple mechanism that allows one entity (application or database) to communicate with another entity or entities. In other words, middleware is any type of software that facilitates communications between two or more software systems. It is able to hide the complexities of the source and target systems, freeing developers from focusing on low-level application programming interfaces (APIs) and network protocols, allowing them to concentrate on sharing information (Kramp & Coulson, 2000).

It is noted that middleware are evolving, which means that they may extend their functions beyond their originally designed functions. In the following, some well known middleware types are discussed (Linthicum, 2001):

- ***Remote procedure call:*** Remote procedure calls provide developers with the ability to invoke a function within one program and have that function execute within another program on a remote machine. The fact that it is actually being carried out on a remote computer is hidden. Remote procedure calls require more bandwidth than other types of middleware products because carrying out a remote procedure call requires so much “overhead.”

- ***Message-oriented middleware:*** This is a queuing software, using messages, which are byte sized units of information that move between applications, as a mechanism to move information from point to point. Because message-oriented middleware uses the notion of messages to communicate between applications, direct coupling with the middleware mechanism and the application is not required. Message-oriented middleware products rely on an asynchronous paradigm. Message-oriented middleware typically provides a structure (a schema) and content (data) in accord with the schema and its use of messages is relatively easy to manage.

- ***Distributed objects:*** Distributed objects are small application programs that utilize standard interfaces and protocols to communicate with one another. Two types of distributed objects are very popular today: common object request broker architecture (CORBATM) and component object model (COMTM). CORBATM and COMTM provide specifications that outline the rules that developers should follow when creating a CORBATM-compliant or COMTM-enabled distributed object. CORBATM is heterogeneous, with CORBATM-compliant distributed objects available on most platforms. COMTM must be considered native to Windows TM operating environments and therefore homogenous.

- ***Database-oriented middleware:*** This facilitates communications with a database, whether from an application or between databases. Database-oriented middleware are of two basic types: command line interfaces and native database middleware. An example of command line interface is Microsoft's open database connectivity (ODBCTM). It exposes a single interface in order to facilitate access to different databases and uses drivers to accommodate differences between databases. Native database middleware accesses the features and functions of a particular database, using only native mechanisms.

- ***Transaction-oriented middleware:*** Transaction middleware does a commendable job of coordinating information movement and method sharing between many different resources. Although it provides an excellent mechanism for method sharing, it is not as effective as simple information sharing.

- ***Transaction Processing monitors:*** They are based on the concept of a transaction, a unit of work with a beginning and end. The reasoning is that if the application logic is encapsulated within a transaction, then the transaction is either completed or rolled back completely. The load-balancing mechanisms of transaction processing monitors guarantee that no single process takes on an excessive load.

- ***Application servers:*** They provide not only for the sharing and processing of application logic but also for connecting to back-end resources including databases, ERP applications, and even traditional mainframe applications. They also provide user interface development mechanisms and mechanisms to deploy the application to the platform of the web.

- ***Message broker:*** This facilitates information movement between two or more resources and can account for differences in application semantics and platforms. They can transform the schema and content of the information as it flows between various applications and databases.

2.11.3. Adapters

Adapters play a key role in application integration. There are two different types of adapters existing on the market today. The first type of adapter is more or less like a connector. It is a basic communication interface into or out of a particular system or database. It also connects the application to the middleware products. In this case, only the middleware has the application logic in it to integrate different applications. The second type of adapter has the application logic in it to integrate different applications. Consequently, the second type acts more or less like middleware. Both types of adapters are explained in detail below.

- ***Adapters with application logic:*** These adapters deliver all-important business logic at the application sub-module level. Any adapter without application logic, combined with many lines of hard code, can get data from SAPTM manufacturing into an OracleTM financial application, for instance. In contrast, an intelligent adapter should be able to insert the data into the proper accounts payable field or invoice line item and meet any pre-determined business requirements when doing so. This requires application-level logic to perform the necessary actions (Traverse, 2001). The best adapter solutions include an intuitive, template-based adapter development environment. They reduce the need for low-level programming and let customers build or modify adapters in a graphical, drag-and-drop environment.

- ***Adapters without application logic:*** This type of adapter acts just like a connector. The middleware has the knowledge about where the data would be and what it means semantically. The adapter simply converts the data from its native format to the format compatible with the middleware. Some of the adapters of this type are called "Universal Adapters." They provide middleware-independent transformation and universal adapter development tools. Figure 2.22 shows the architecture of a universal adapter (eJai, 2001).

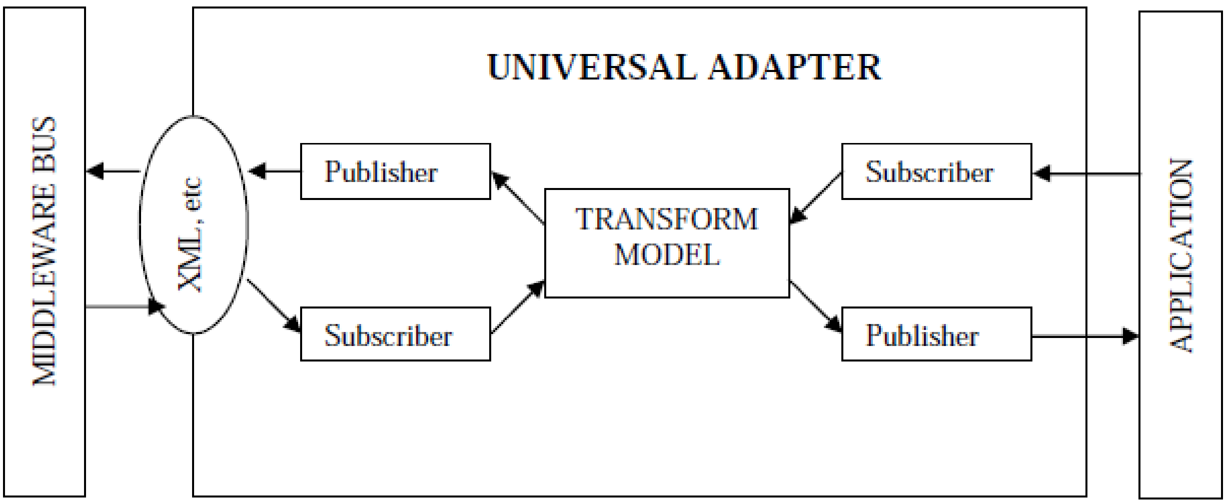


Figure 2.22 Universal Adapter (Vernadat, 1999)

Universal adapters have transformation models inside them that convert data from the native application format to the format required by the middleware. They handle publishing and subscribing in addition to reply/request functions to enable application formats usable by middleware (eJai, 2001).

2.12 System Complexity

Today's information systems show a continuous growth in complexity. This growth was identified by Manny M. Lehman back in 1985, who placed systems into certain classifications. These were 'specification systems' (where a complete specification of the system is laid down and must then be fulfilled by the implementation, eg assigning car-plates); 'problem systems' (where only a limited, incomplete knowledge of the problem and the resulting specification is available and the implementation is challenged to find acceptable solutions, eg a traffic simulation program); and 'environment systems' (which have the same properties as problem systems with the added difficulty that their very introduction into the system environment will change the problem, eg control of traffic lights based on simulation with the burden of the uncertainty of drivers' future behaviour) (Lehman, 1985). Hermann (2003) added a fourth category - 'wicked problems' - which have the additional property of being large, complex, ill-defined and lacking a clearly identified objective (eg asking for the traffic control to also optimize the travel times of cars). He observed that such a system cannot be specified without some prior concept of its solution. Changing these concepts at a later stage cannot be done without considerable loss of time and effort. As a consequence, ill-conceived projects must either be continued with an inappropriate basic architecture or be abolished.

Contemporary businesses have of necessity installed IT infrastructure in order to operate, compete and grow in the current business climate. Highly abstracted, enterprises can be represented as the complex interplay of people, processes and technologies in achieving business objectives. These objectives are typically articulated as financial, operational and market success measures. While IT infrastructure has obviously accelerated the pace, reach and agility of business, it has also introduced a huge amount of complexity into enterprise fabrics. Typically, roughly 80 percent of an IT budget is spent on maintenance and development of existing enterprise systems while 10 percent of the entire workforce is devoted to IT operations; in addition, associated costs are difficult to control and maintain. Therefore, a radical shift is necessary whose principal goal is "to kill the complexity." Whichever road we take, we must cope with huge volumes, rising dynamics, conflicting constraints and self-defeating uncertainty. These are the principal contributors to enterprise complexity.

2.12.1. Complexity Factors and Causes

Historical developments created at least three sediments of computing systems: mainframes, server-client systems and Web systems. Each had very specific creation factors and driving causes. Only recently has it become clear that integration and interoperations are key for the flexible, evolving and reliable enterprise architecture consisting of these sediments. Competition among vendors and inability to agree or impose common standards led to heterogeneous systems that talk to each other only through adapters and operational bridges. Eventually, this stimulated the rise of enterprise middleware systems whose only function is to mediate among incompatible systems. Development style, budget pressure and project deliverables followed by the necessity to deliver short-term results in an over-constrained environment led to the rapid creation of huge amounts of proprietary code. Maintenance and operation of this "mountain-of-spaghetti" code base represent a logistic nightmare. The volume of enterprise data has risen sharply also, as productivity tools enabled enterprise employees to create quickly huge numbers of big documents leading consequently to lower content quality. This was followed by sharpened expectations about responsiveness and latency within and out of the enterprise. All of the factors mentioned above contribute to the huge growth in enterprise IT complexity.

2.13. Concept of Enterprise Integration

Enterprise integration (EI) is the re-engineering of business processes and information systems to improve teamwork and coordination across organizational boundaries, thereby increasing the effectiveness of the enterprise as a whole. EI is enabled by interoperating enterprise models

Enterprise Integration aims to connect and combines people, processes, systems, and technologies to ensure that the right people and the right processes have the right information and the right resources at the right time. It enables successful operation, in a world of continuous and largely unpredictable change, of a single manufacturing company or an ever-changing set of extended or virtual enterprises, by enabling quick and accurate decisions and adaptation of operations to respond to emerging threats and opportunities, (Brosey et al., 2001). Below is the enterprise integration model;

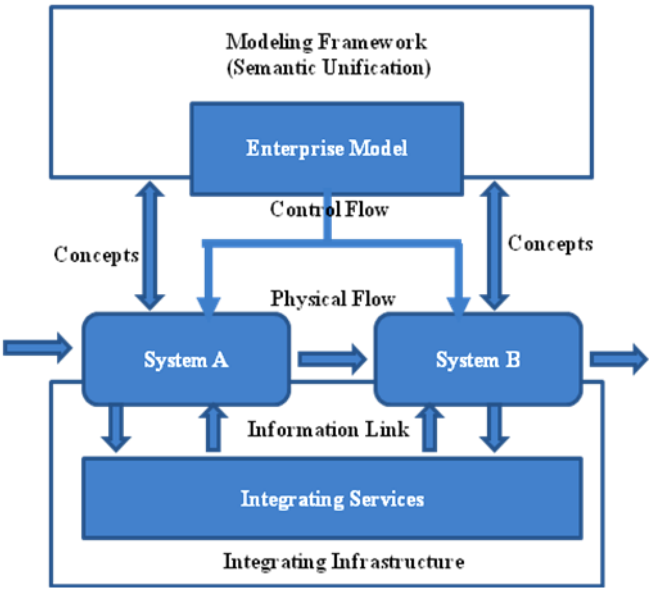


Figure 2.23 Enterprise Integration Model (Brosey et al., 2001)

2.13.1. The Need for Enterprise Integration

To maintain and improve their 'competitiveness' while reacting to the fast 'changes' that take place in the opportunities and needs of the market, enterprises must adopt a form of organisation and operations that will allow them to obtain the 'maximum benefit' from their resources.

Recently, several approaches oriented toward the improvement of the enterprise's competitiveness have appeared. (McCarty, 1993) listed the main 'tendencies' which emphasize total quality management, process re-engineering, collaboration between enterprises, virtual enterprise, improvement of the availability of information, flexibility and the integration of clients and suppliers. These new tendencies and innovations in the field of management and technology have almost always been handled in the enterprises in an isolated and uncoordinated way. Thus the large promised 'improvement expectations have not been accomplished'.

Therefore, in order to achieve all the possibilities that these new and better methods and tools offer, an enterprise must 'efficiently manage' all its elements, 'aligning' and 'integrating' them in order to improve the ability to work together in a 'continuous improvement process' toward the accomplishment of the objectives and the strategy of the enterprise.

The necessity for 'integration in the enterprise' has been developed by many different authors like Amice (1993) and Neuscheler & Spark (1994) amongst others. One of the definitions is that of (Lawrence & Lorsch, 1968), who defined integration as "the quality of the collaboration that exists between departments (marketing, finances, manufacturing, store, etc.) in order to 'satisfy' the environmental requests and the processes, methods and resources thanks to which it is satisfied". The work developed by (Petrie, 1992) is also interesting, as it analyzed and justified integration as the key variable to support new requirements of the market. In this way, the conclusion of the work carried out by the National Research Council of USA, when asked by NASA, on the 'impact that the efforts toward integration' were producing in concrete American enterprises like Westinghouse or General Motors (National Research Council, 1986), allowed us to determine that the benefits of integration are of two types: 'strategic benefits' (which give a competitive advantage to the enterprise, establishing the best business objectives and adequate means to reach them) and 'tangible and quantitative benefits'. These latter results in:

(1) Decreasing delivery times of the products (30 – 60%)
(2) Product design time (20 – 50%)
(3) Product design cost (15 – 30%)
(4) Manufacturing mean time (50 – 70%)
(5) Space (25 – 40%)
(6) Inventory cost of the raw materials (30 – 60%)
(7) Product being produced (60-80%)
(8) Direct manpower cost (20-35%)
(9) Indirect manpower cost (30-35%)
(10) Quality cost (25-40%)
(11) Throughput increasing (40-70%)

The needs of enterprise integration may arise from many perspectives (Nell & Kurt, 1997), such as:

- **Identify the right information:** This requires a precise knowledge of the information needed and created by the different activities in the enterprise operation. Knowledge is structured in the form of an accurate model of the enterprise operation which describes product and administrative information, resources and organisational aspects of the operational processes and allows what-if analysis in order to optimize these processes.

- **Provide the right information at the right place:** This requires information sharing systems and integration platforms capable of handling information transaction across heterogeneous environments. Environment which consists of heterogeneous hardware, different operating systems and monolithic software applications (legacy systems). Environments which cross organizational boundaries and link the operation of different organisations on a temporal basis and with short set-up times and limited time horizon (extended and virtual enterprises).
- **Integration of markets:** Enterprises adapt regional product consumptions and servicing as a response to the new free trade economic areas that are established around the world.
- **Integration between several development and manufacturing sites:** This is the collaboration between remote enterprises that is needed to produce complex products.
- **Integration of multivendor software and hardware components:** Systems need to integrate with various IT hardware and software solutions (Vernadat, 2002).
- **Up-date the information in real time to reflect the actual state of the enterprise operation:** This requires not only the up-date of the operational data (information created during the operation), but adapting to environmental changes as well.
- **Co-ordinate business processes:** This requires precise modeling of the enterprise operation in terms of business processes, their relations with each other, with information, resources and organization. This goes far beyond exchange of information and information sharing. It takes into account decisional capabilities and know-how within the enterprise for real time decision support and evaluation of operational alternatives.

2.13.2. Evolution in the Enterprise Integration Concept

The ‘enterprise integration’ concept, as it is posed in this research work, is a new concept that comes out of the evolution of the influences of different focuses such as business re-engineering, total quality management, integrations of customers and suppliers, ERP’s etc.

The first approach toward industry integration was focused mainly on the technological aspects, solving connection problems between different devices and exchange of information between computer applications. However, in the new action framework of the enterprise, it is necessary to adopt a more global focus, in agreement with the definition of 'enterprise integration', which should cover the whole enterprise and adopt an organizational focus, more than a technological one. Today, enterprise integration is applied to any type of enterprise (industrial, service, transport, businesses, etc.), and one of the main current lines of research in this field is the integration of enterprises from different sectors that play a role in the lifecycle of a product or service (virtual enterprise). Figure 2.24 shows the various levels of integration.

	Resource/ Integration Need	**Examples of Integration Mechanisms**	**Enabling environment /Infrastructure**	
Organizational Integration	Organizational Units (Functions/Departments)	E-mail, collaborative software, lateral teams ----------------------------------- Top Management Strategy, budgets, performance metrics	Organization policies/ structure	
	Decision Makers	Email, collaborative software, knowledge management systems ----------------------------------- Face-to-face meetings, job design, performance metrics		
Systems Integration	Business Processes (both internal & external to the firm)	Workflow, Collaborative Systems, SCM, CRM, Web Services ----------------------------------- Process owners, teams, performance metrics, service level agreements	Standards	Systems Architecture
	Applications	Inter-process communication, RPC, Messaging, ERP, Web Services	Networks	
	Data	Data Dictionaries Databases, XML	Platforms	

Figure 2.24 Framework for Enterprise Integration (Linthicum, 2003)

2.13.3. Enterprise Integration and Information Systems

It is important to clarify in this point the difference between an integrated enterprise and an integrated information system. 'Integration of an enterprise consists of putting components together to form a synergistic whole. It means integration on a grand scale, a scale that transcends traditional external and internal corporate boundaries. Enterprise internetworking using electronic networking to form close ties with suppliers, distributors and customers. Problem solving and decision making are conducted by flexible teams cutting across the individual enterprises and distributed over time and space. It is a combination of horizontal integration for a better control of material and information flow and a vertical integration for efficient control of the decision flow. In essence, Enterprise Integration is concerned with facilitating information, control and material flow across organizational boundaries. This is done by connecting all necessary functions and heterogeneous functional entities in order to improve communication, cooperation, and coordination within the enterprise so that the enterprise behaves as an integrated whole, therefore enhancing its overall productivity, flexibility and capacity for management of change'. **Integrated Information System** is practically an expansion of a fundamental information system achieved through system design of an improved or broader capability by functionality or technically relating two or more information systems or by incorporating a portion of the functional or technical elements of one or more information systems to another. The main objective of integrated information system is the total elimination of stovepipes in operations and information systems.

To achieve this goal it is necessary to adopt an innovating viewpoint regarding information. This should be considered a fundamental mechanism for the total integration of the enterprise's engineering and management functions. From this approach, a fundamental objective for any enterprise integration project is the need to create a 'global information infrastructure' supported by the new information technologies. This should be an infrastructure focused on flexibility and efficiency in its functioning. It should (1) carry out efficient information processing offering correct information at the appropriate time; (2) allow for the co-operation between the enterprise's subsystems and its external elements; (3) cover up the heterogeneity of physical resources and information application and (4) be able to respond to changes in the enterprise's way of functioning and the evolution of support technologies (Mayer & Painter, 1991). Nevertheless, an integrated information system does not assume that enterprise activity has been integrated. Even though the incompatibilities that impede physical communication and exchange of data and information have been overcome, enterprise decisions may still be made in order to optimize particular and contradictory objectives. Only when the information is used within the ‘co-operative integrated management framework in order to ‘co-ordinate’ activities and decisions toward strategic objectives ensuring communication and collaboration among groups and individuals in the firm, will the information behave as a valid mechanism for ‘total enterprise integration’.

To build a computerized integrated information systems capable of providing correct information wherever it may be needed, incompatibilities appearing on both the physical level between different communication networks and protocols and the logic level between databases and software, must be overcome.

Of these two elements, software proves to be the most restricting. There are two alternatives for obtaining software adapted to a particular enterprise’s needs: to develop customized software or to use a standard solution. Each alternative has its advantages and drawbacks.

Theoretically, a customized development will always be closer to the enterprise’s particular needs. Even so, cost and time are considerable. Moreover, inevitable changes in the system (both physical and logical) may require an almost complete overhaul of applications thereby wasting large part of previous investments.

The other option is to use a manufacturer developed standard solution. This solution is parameterized to enterprise’s needs trying to limit changes to the original. In this way, it is ensured that future versions and improvements that the manufacturer may include in the software can be adapted to the enterprise. Two types of standard solution applications can be distinguished, namely sectorial software and ERP systems.

Sectorial software set is a set of computer applications developed in order to fulfill the particular needs of a sector. The software enterprise tends to be a small or middle-sized computer firm geographically situated near a cluster of enterprises belonging to the same sector and which posses a profound knowledge of its needs. As a result, the software is well adapted to this kind of enterprise thus implementation time is short and due to scaled economics, the cost is not excessively high. Its main disadvantage is the size of the software enterprise which makes it unable to possess enough resources to extend the product in all areas of the enterprise and to ensure technological updating. The closed systems are very specific, of limited flexibility and do not deal with information as an enterprise resource. This makes total integration and reaping its subsequent benefits impossible to obtain.

The other option is the enterprise resource planning (ERP) systems. These are systems which were first developed to cover the needs of one sector. However, thanks to the widespread of ERP use, the developers have turned into large multinational enterprises and their functions have been extended to include a large number of enterprise activities. Examples of these are SAP, BAAN, ROSS, and PeopleSoft. Its strength include (1) size and stability of the software enterprise, which guarantees constant technological updating; (2) the database and computer applications are integrated which enables information to be shared; (3) it involves a large part of the enterprise and (4) some ERPs have process modelling and documentation tools which in some way, allow to join the organization, the software and quality assurance.

Nevertheless, the time and the cost of every implementation are too high, especially when modifications to the standard application are necessary. To avoid this, manufacturers tend to develop Sectorial maps. Basically, this consists of adding a layer over the standard software (without modifying the kernel of the ERP) closer to the needs of the sector. In this way, there will be far fewer modifications when time comes to implant it into a particular enterprise. Nevertheless, the function of even the most widespread and advanced ERP system is not the same and this should be taken into account. This is due to the fact that having been initially developed for a particular sector; they are always better-prepared for this sector and less prepared for other sectors than their competitors. The current trend is to widen their performance realm to cover not only operative and management type processes but also to offer information for strategic decision-making and connection among enterprises, thus giving solutions for e-business (Figure 2.25).

From the standpoint of enterprise integration, ERP systems seem to be the best solution currently. Appropriate software enables the enterprise to improve efficiency though sometimes forces people to work in a certain way. This is one of the main reasons of delays in software implementation. The managers of the enterprise think that software is like a machine, which you buy and begin to work quickly. They do not see software as a complex element which affects the entire organization and functionality and which interacts with a large part of the firm’s human resources.

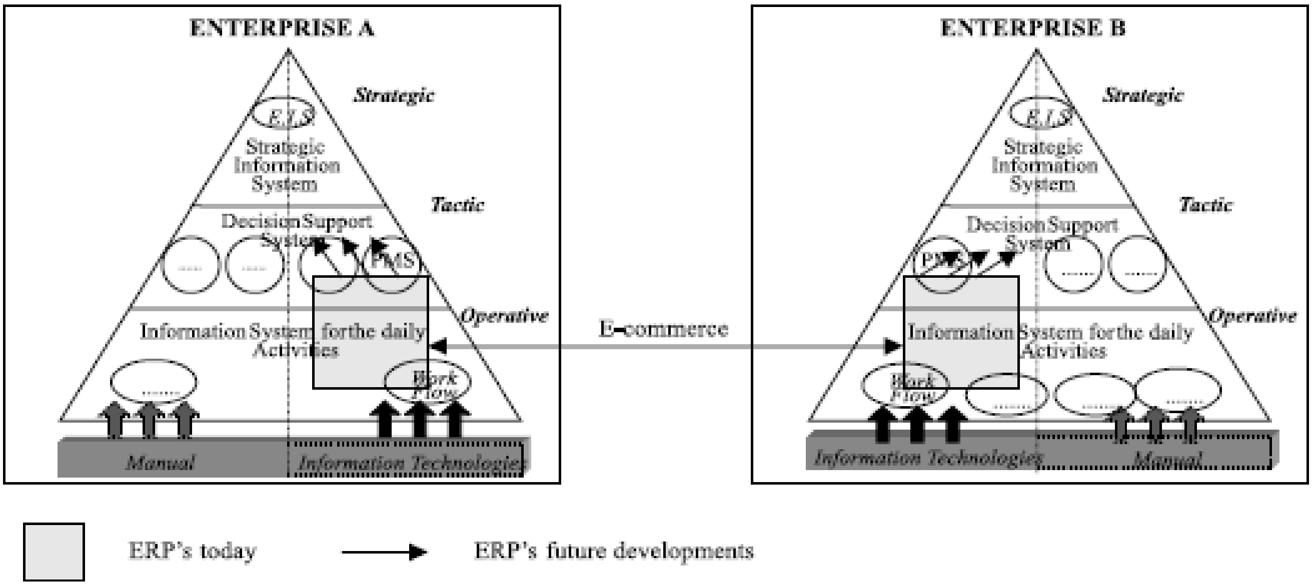


Figure 2.25 Mapping the EIS's position in the Enterprise (Mayer & Painter 1991)

2.13.4. Problems with Enterprise Integration

At this time, there exist a large number of enterprise integration architecture such as CIMOSA, PERA and RM-ODP, to name only a few (Williams et al., 1993; Katzy et al., 1993; Doumeingts et al., 1993). The problem with any enterprise integration architecture is that a large part of the success of the integration project is not determined by the architecture used but the manner in which it is implemented. The success of an integration project can be said to be determined by internal organizational philosophies rather than a specific architecture. In particular, for an integration project to succeed, it is necessary to manage the total information resource of the organisation as a single entity. It is for this reason that we defined a model that specifically demonstrates the relationship and interaction between all the information sources with an organisation. Some other problems with enterprise integration include the following:

- Systems for manufacturing and enterprise integration cannot be bought off the shelf; each organisation needs to develop its own integrated information system.
- The necessary expertise to develop an infrastructure for enterprise integration requires that experts of different disciplines cooperate closely.

- Enterprise integration projects are usually initiated in existing organisation in which substantial amounts are already invested in systems that might not support integration.
- The risk involved in an enterprise integration project can be high.

In examining the problems associated with integration projects, it is noteworthy that none of these problems are caused by lack of technology. In fact, the problems are caused because technology developed faster than the ability of large organisations to use it optimally. In other words, the problem is not lack of technology but the way in which we apply it.

CHAPTER THREE
METHODOLOGY AND SYSTEMS ANALYSIS

3.0 Introduction

System analysis is the study of a problem prior to designing a solution for it. The aim is to build a problem model, that is, to create a description of what is required and what will eventually be built without attempting to say how it will be built.

3.1 Methodology

Methodology is the study of the various system analysis and design methodologies. The purpose for all system analysis is to find what information the system should manage, what fact to find and search for, and how to find them. However, to execute these objectives, Object-Oriented Analysis and Design Methodology (OOADM) with Unified Modeling Language (UML) has been adopted in the analysis and the design of the proposed model and the case study application. UML is the standard language for modeling large-scale software systems. UML includes several types of diagrams in order to model the static and dynamic behavior of a system. The OOADM specifies the modules, stages and tasks which have to be carried out in advance and the deliverable to be produced.

3.1.1 Method of Data Collection

This refers to the various methods used by the researcher in the analytical study of the existing enterprise architectures in order to find the various components of the model and their development methodology. These methods are listed below;

i. Theoretical research (literature review, internet and library)
ii. Review of Documentations
iii. Interview of software developers

i. Theoretical Research

These are ways or instruments used in obtaining more information. The internet often provides vast and current information relating to the topic under review. Most often the library research method is considered as one of the leading roles in research methodology.

The literature provided the researcher with a theoretical background. The materials reviewed included published journal articles, conference proceedings, book chapters, and websites in the fields of IS/IT-Business strategy alignment, information system architecture, information architecture, information technology architecture, Enterprise Architecture and Enterprise Architecture Frameworks. After categorizing the literature, the researcher then highlighted important points for each piece of work, synthesized and summarized the major ideas that surfaced among each category. These ideas include what information an enterprise needs to deal with change, how EA is used to implement change, the various representation of the enterprise architecture components and their attributes. The literature review embodied the basis for the design of the framework.

ii. Review of Enterprise Architecture Documentations

To further explore how the theoretical work is applied in practice and to make sure that the construction of the framework works in practice, the researcher collected documentations of the usage of Enterprise architecture in real situations, such as Enterprise Architecture Method Handbooks. Documentations gave the researcher a thoughtful and comprehensive representation of the situations.

iii. Interviews with IS Developers

The conduct of interview normally involves the participation of at least two (2) persons. The interviewer and interviewee. In this method, there has to be a mutual understanding between them so as not to divert from the actual area of research. Some information system developers were interviewed on the current practice on information system development in the enterprise and their view on enterprise architecture. The result showed that they value the use of enterprise architecture but are yet to really understand the elements as it is a new paradigm in software engineering yet to be studied in this part of the world.

3.2. System Analysis

Enterprise architecture (EA) is a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and IT infrastructure. The review of relevant literatures indicates the need for EA in small and medium-sized enterprises (SMEs), important drivers of the economy, as they struggle with problems related to a lack of structure and overview of their business. However, existing EA frameworks are perceived as too complex and, to date, none of the EA approaches are sufficiently adapted to the SME context. Although EA might offer SMEs a solution to typical problems related to a lack of overview, strategic awareness, IT planning, and business-IT alignment, EA approaches that cater for the specificities of small businesses are still missing.

In a SME, the entrepreneur (CEO) controls the enterprise. However, while most entrepreneurs have a good knowledge about their company, the overview tends to stay unspoken. This can cause some problems to occur as listed in (section 2.8).

From these problems and the SME criteria in 2.8.1, three major areas were identified for improvement which would create value and reduce operation cost in SMEs. These include;

(i) Standardization of business processes.
(ii) Integration of systems.
(iii) Better use of technology to improve process efficiency.

It is obvious that EA could help to reduce these problems; however, this area of research is not yet totally satisfactory because existing EA frameworks and methods are generally unknown and unused concept in SMEs.

3.2.1 Analysis of Existing Enterprise Architecture and Models

Although significant amount of research has been done and many models proposed on enterprise architecture and enterprise architecture frameworks, analysis of literature and documentations shows that there are some drawbacks in the existing methods and EA frameworks and methods are generally unknown and unused concept in SMEs.

. They include;

(a) The **information system architecture** proposed by Zachman (Zachman, 1987) (See 2.6.2.1) offered a systematic taxonomy of concepts for relating the components that describe and represent the real world system and its implementation. The drawbacks are;

i. It is only a reference framework – has no defined methodology
ii. It is complex and large because of the number of cells

iii. The metaphor used throughout the Zachman Framework is that of the architectural development of a building, or manufactured product and does not adequately suit the reality of information system development.

iv. Independent models: The purpose of a framework is not to fill in all its cells, but a framework should be a logic structure for classifying and organizing the different architectural descriptions of the enterprise. There should be consistency between the models, not just a bunch of independent models;

v. No model for enterprise application classification.

vi. The entities were defined (architectural views) without attributes that contribute to them.

vii. The Zachman's Framework did not suggest how resources are aligned to support corporate goals and strategies;

viii. Has never been adapted for SMEs because of the complex nature.

(b) The Open Group Architecture Framework (TOGAF) (See 2.6.2.4) proposed by the Open Group is a framework for developing enterprise architecture. It is described as a set of documentation published by The Open Group (Open Group, 2002). The drawbacks are;

i. It uses a bottom-up construction focus of the enterprise functional system, defining at the beginning elementary tasks and then gathering them into enterprise entities to satisfy strategic objective.

ii. Current version is excessively IT-Centric.

iii. Business architecture is poorly described.

iv. Over-emphasis on low level technology architecture Entity definition without attributes.

v. Has never been adapted for SMEs because of the complex nature

(c) Federal Enterprise Architecture Frameworks (FEAF) (See 2.6.2.5) proposed by the council of Chief Information Officers of the US Government in the 1990's targets to promote shared development for common federal processes, interoperability and sharing of information among federal agencies and other governmental entities (CIO Council, 2001). The drawbacks are;

i. It is a proprietary framework owned by U.S government to support their operations.
ii. It is only a reference framework – has no defined methodology.
iii. Has never been adapted for SMEs because of the complex nature.

(d). **Architecture for Integrated Information Systems** (ARIS)(See 2.6.2.3) proposed by Professor Scheer at the University of Saarbrucken in Germany (Scheer, 1996), focuses only on the design of enterprise information systems. The disadvantages are;

i. It did not include any design methodology.
ii. ARIS is specifically for information system and did not consider the enterprise and business that owns the system.
iii. Methodology is based on normal system development.
iv. Has never been adapted for SMEs because of the complex nature

3.2.2 Weakness of Existing Models of Architectural Frameworks

i. From the analysis, it can be seen that these architecture models were focused mainly on large enterprises. They did not consider the nature of small and medium enterprises (SME's). It would be interesting to prove these architecture models (EA) usefulness in SMEs.

ii. The existing enterprise architectures lack the approach that can be used by SMEs to develop their EA models and manage their EA.

iii. The existing architectures lacks information about how these frameworks achieve business/IT alignment, manage system complexity and how they identify the concept of business value in SMEs.

iv. Most of the commonly used information systems architecture approaches has not been able to identify scope and fully define the boundaries of its information systems. This in turn, leads to a high level of data redundancy and poor quality information.

v. The existing systems/architectures did not offer a complete solution that models all the characteristics of an enterprise system in an SME.

vi. The Infrastructure model is based on technologies and hardware without considering Liveware (people). Also power, energy and security problems were not considered as it is obtained in developing countries like Nigeria.

vii. Most of the models did not consider evolution and improvement in their methodology.

viii. The existing models failed to fully capture the unique characteristics of enterprise system in SME, that is, they support poor IT - business alignment in SME.

ix. The systems developed using most of the architectures are complex in nature based on the fact that the architecture is complex.

x. None of the models have been adapted in EA for SME context.

The result shows that a lot of research has been done on EA field, still hardly anything is known about its use in the context of a small and medium sized enterprise (SME). Because of some specific characteristics of SMEs, however it will be remarkable to apply EA to an SME.

3.2.3 Feasibility Study

During this process, investigations were thoroughly made in order to develop the new architecture model for SMEs (Integrated Model) in sufficient depth. Based on the review made of different existing architectures, two types of architectures were identified with different objectives. These are (i) The ones directed toward the development of 'integrated information system' and (ii) The ones which cover the 'whole enterprise integration project'. The latter are more oriented toward the problem this dissertation is focused. The ‘reference architecture (RA) for integration of information’ has the objective of developing an integrated information infrastructure that communicates and coordinates the different technological devices that generate, process, distribute and supply information. One of the most well known is the ARIS Architecture. The ‘reference architectures for enterprise integration’ are oriented towards the integration of the whole enterprise, considering not only the technological aspect of the system but also the economic, social and human ones. Among the most well known ones are CIMOSA, TOGAF, and FEAF. Based on the information assessment (what is required), information collection, it was observed and agreed that the implementation of an integrated framework for enterprise system will help resolve the problems of enterprise IT systems complexity, achieve better alignment of information technology with business objectives and realize real business value from the implementation of the system. The integrated model for small and medium scaled enterprise system was designed as an instantiation of enterprise architecture in terms of a set of software modules, computer platforms, network components, and databases assembled in such a way as to be able to process business transactions and thus meet all the system requirements specified in that Enterprise Architecture and provide a methodology for continuous evolution based on Continuous improvement process.

3.3 Analysis of the Proposed Model: An Integrated Model for Small and Medium-Scale Enterprise Systems (AIMES)

Our solution consists of developing a new EA approach guided by the requirements for EA in an SME context (cf. section 2.8). The approach was called AIMES, so that these requirements would always be kept in mind.

The AIMES architecture model comprises five architecture components: Business Architecture, Information Architecture, Information system Architecture, Infrastructure and Resource architecture and Security and Governance Architecture. Each of this sub-architecture is individually represented and organized as a UML package as depicted in Figure 3.1. Each package owns its model elements and its elements cannot be owned by more than one package. The relationships depicted with arrows represent the dependencies of each package.

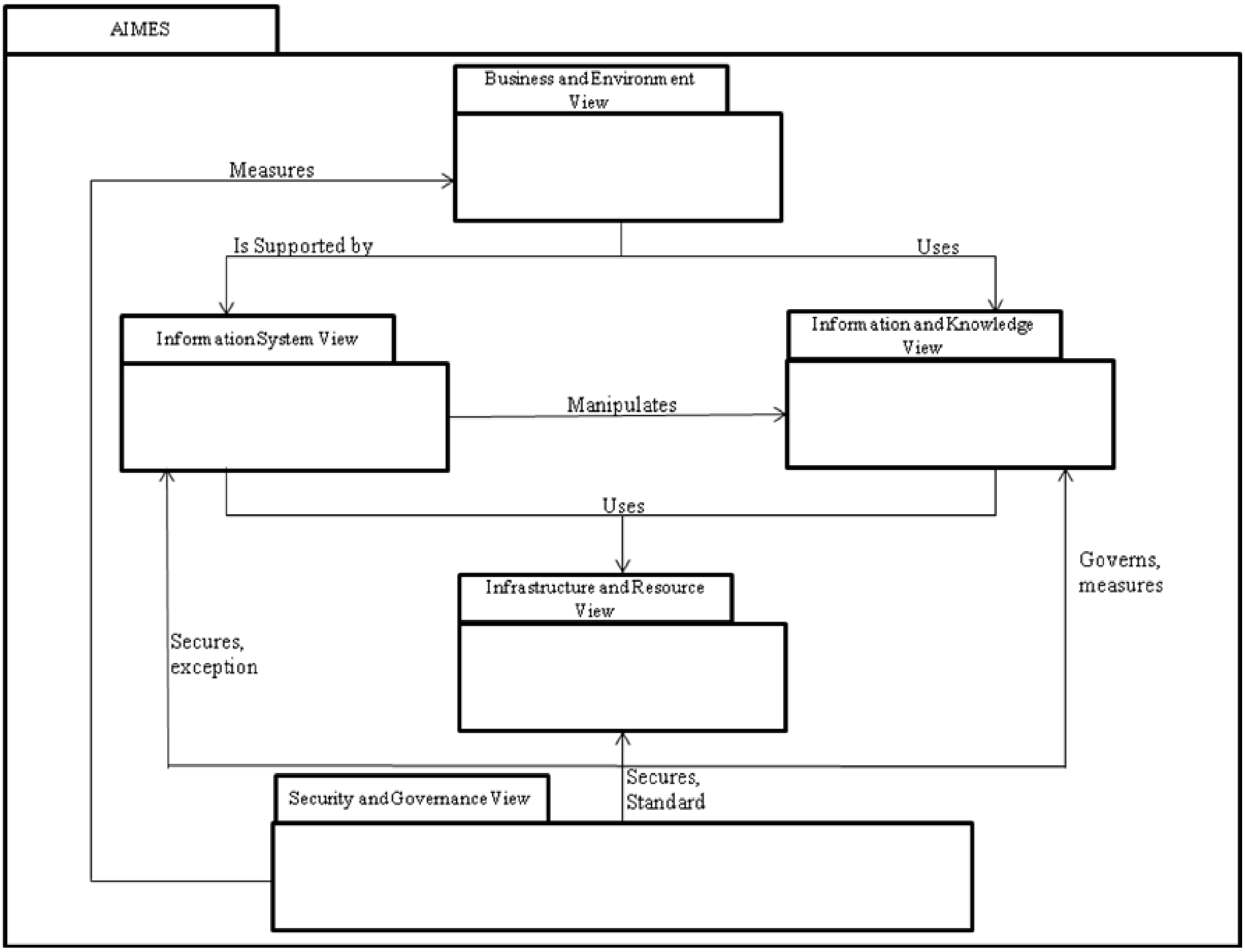


Figure 3.1. Conceptual AIMES Metamodel Showing the Five Views and Interdependencies

We propose modeling the multi-dimensional aspects of AIMES and based on that, defining and evaluating the alignment between business processes, business information and the corresponding supporting systems and infrastructure. The first step in this direction is identifying a minimal set of components able to represent the required organisational concepts while ensuring that the alignment between these concepts can be accessed. Figure 3.1 is a conceptual metamodel showing the interrelationship between the five AIMES components.

3.3.1 AIMES Use Case Description

Table 3.1 AIMES Use Case

Model Type	Description	Tool	Actors Involved
Business model	The business model results from the implementation of business strategies and definition of processes	Organisation charts	Enterprise Architect, Business Analyst
Information Model	The information model describes what the enterprise needs to know to run its processes and operations as described in the business model.	Data flow Diagram, ERD	Information Architect, Enterprise Architect
Application Model	The information system defines the applications needed for data management and business support regardless of the actual software used to implement systems	Enterprise Architect	System Architect, Enterprise Architect
Infrastructure and Resource Model	The infrastructure represents the technologies behind application implementation as well as the infrastructure and environment required for the deployment of the business process support systems	Network engineer	System Engineer, Enterprise Architect
Security and Governance Model	The security and governance represents the general security, monitoring and management of standards for implementation of the entire model	Solution Engineer	System Architect, Enterprise Architect

(i) AIMES Use Case Diagram

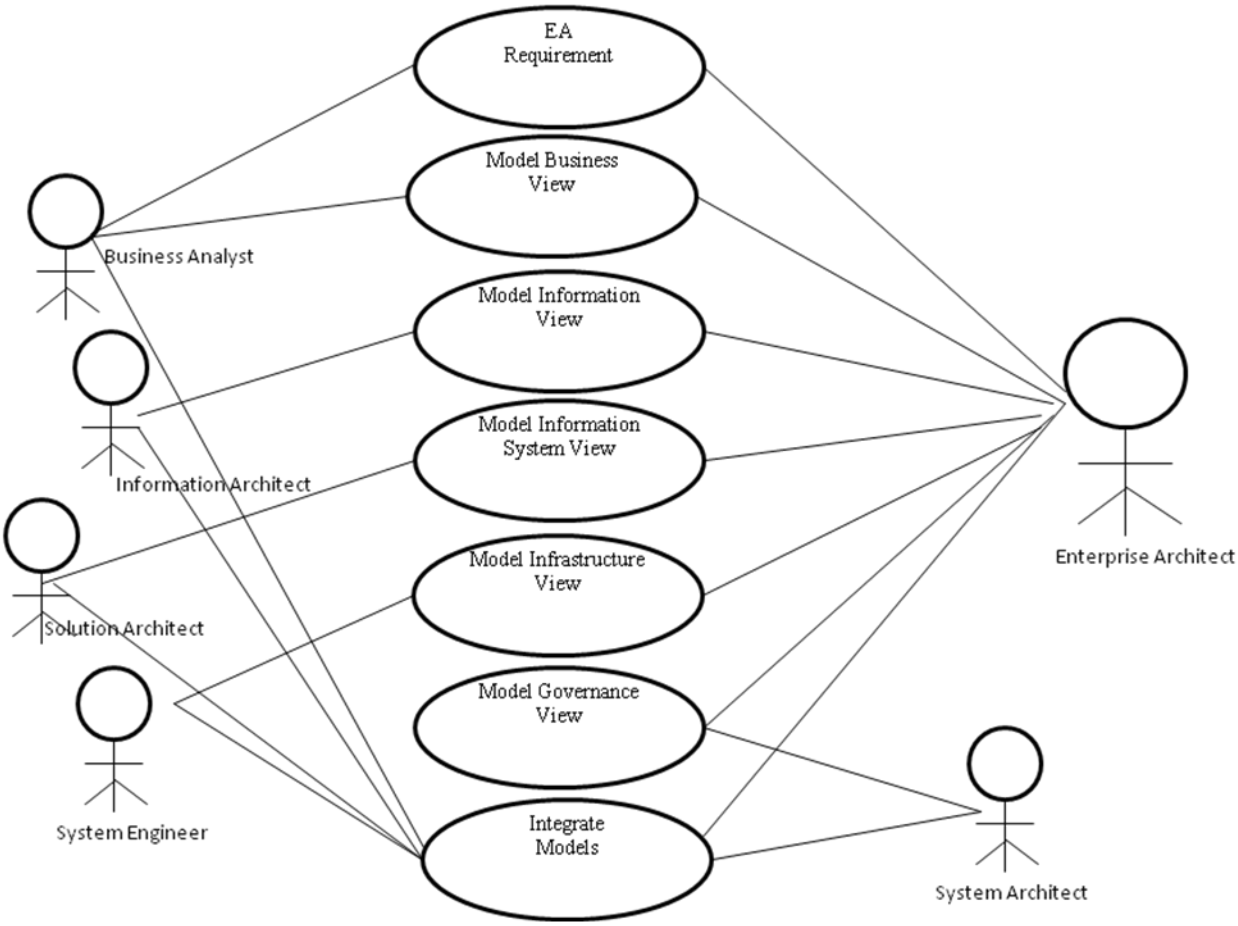


Figure 3.2 AIMES Use Case Diagram Showing Link between Stakeholders and the Views

(ii). AIMES Use Case Description

AIMES addresses one or more concerns of the enterprise stakeholders such as the business, information view, information system view, infrastructure and resource view and governance and security view. In order to create a useable and maintainable enterprise architecture model, different actors and stakeholders are involved. They aim to achieve a particular view and also define the principles guiding the view. The actors and stakeholders for AIMES are;

i. Business Analyst: Works with the Enterprise architect in the EA requirement and Business view to;

a) Define the main business goals and business drivers of the enterprise.
b) Determine the way in which the business drivers and business goals are realized.
c) Elaborate the requirements identified.
d) Determine the business processes that are related to the operations.

e) Determine the business processes that are related to control of the operations.
f) Map the requirements defined in the in the EA requirements to business processes.
g) Define sub-processes when a business process consists of multiple activities.
h) Define the input and output data for every activity within a process.

ii. **Information Architect:** Works with the Enterprise architect in the information view to;
a) Determine data structure for all data objects identified in the business requirements.
b) Link all data elements within the data structure to elements defined in business requirements
c) Combine data structures into one complete data structure

iii. **Solution Architect:** Works with the Enterprise Architect in the information system view to;
a) Determine which activities or sub-processes will be performed by software application.
b) Define the functions that should be performed as an Application function.
c) Decide what application functions are combined into an application (component).

iv. **System Engineer:** Works with the enterprise architect in the Infrastructure view to;
a) Determine the best technologies for the implementation of the information system.
b) Determine the best technologies for the implementation of the data entities
c) Assess the consequences and prerequisites of technology for business processes and cross-company cooperation.

v. **System Architect:** works with enterprise architect in the governance and security to;
a) Decide the architecture implementation and management strategies
b) Decide on the architectural controls to be enforced.
c) Define the security levels that will be applied to various levels of management.

vi. **Enterprise Architect:** Works with business analyst, information architect, solution architect, system engineer and system architect to;

   a) Integrate the various views defined and designed into a functional and usable model.
   b) For continuous improvement of the architecture.
   c) Evolve or maintain the architecture.

3.3.2 The AIMES Architecture Model for SMEs

The AIMES architecture model was developed with a strong orientation toward its practical application to small and medium enterprises (SME's). This is because the existing architectures are based on large enterprises and large information system projects and also with the Nigeria environment as a core consideration which majorly has small and medium enterprises. Our approach differs from the previously mentioned EA approaches in that the approach is specifically designed taking into account the characteristics of SMEs and their CEO. As Lankhorst (2009) mentioned, it is necessary to use an EA approach that is understood by all those involved from the different domains. SMEs have characteristics, some which are indeed different from larger companies. (Bernaert and Poels 2011b)

3.3.3 Characteristics of AIMES

Among the main characteristics of the AIMES architecture, it can be shown that:

1. It proposes an enterprise vision oriented architecture toward **Business processes.**
2. It establishes the life cycle of the enterprise system based on a ***continuous improvement process.***
3. It uses modeling and simulation as a tool to analyze the decision impact
4. It describes enterprise system as the instantiation of enterprise architecture in terms of a set of software modules, computer platforms, network components, and databases assembled in such a way as to be able to process business transactions and thus meet all the system requirements specified in that Enterprise Architecture and provide a methodology for continuous evolution based on Continuous improvement process.

The architecture uses the *'business process'* as the structural *unit (subsystem) on which the integrated enterprise system is developed.* This methodology allows organizing all the main enterprise elements (activities, resources, information and decisions) in order to maximize the value of the product and services that the enterprise offers and that the clients require.

For the graphical representation of every process, different modeling tools are used inside AIMES. The Event-Process chain will be used for the operative tasks. In addition, the object oriented design and unified modeling language (UML) allows the development of a 'dynamic enterprise integrated model', including all aspects normally modeled separately (functional, decisional, informational and resources). The model can be used to simulate and evaluate the business process performance. The **AIMES** architecture model is organized in '*five views'* (see Figure 3.2).

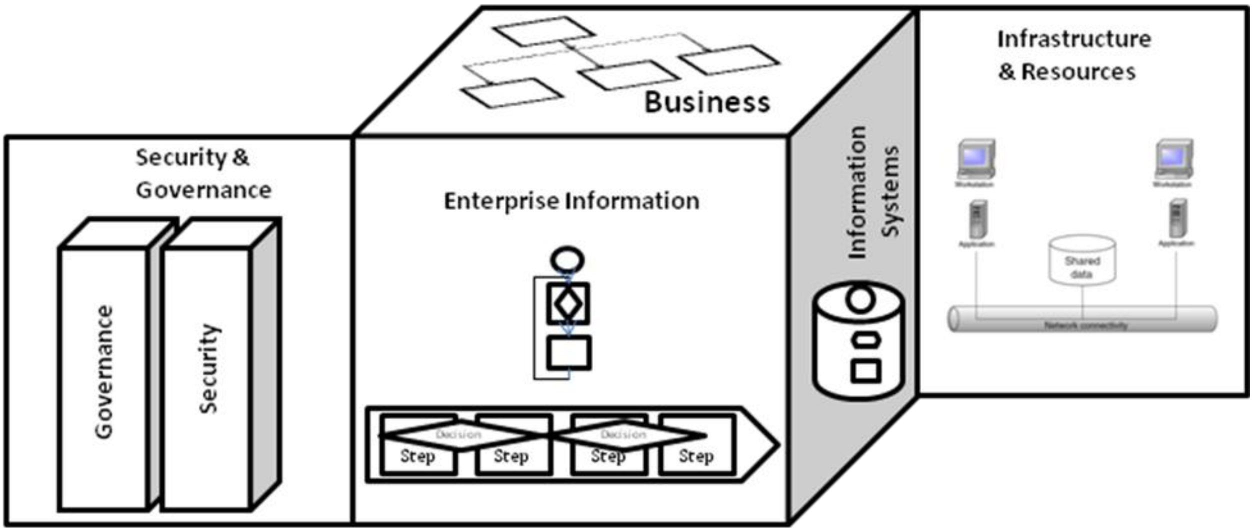


Figure 3.3 Proposed AIMES Architecture Model showing the five (5) Views

3.3.4 Definition of AIMES Views

1. **Business View:** This view answers the question of 'What business are we enabling?' The Business view aligns an organization's operating model, strategies, and objectives with IT; it also creates a business case for IT transformations and provides a business-centric view of the enterprise from a functional perspective. This part of the framework provides the following three key areas of information about the business:

(i). **Business Strategy**: Key business requirements, objectives, strategies, key performance indicators, business risks, and the business-operating model (how processes and systems are centralized versus decentralized across the business).

(ii). **Business Function**: The key business services, processes, and capabilities that will be affected by the enterprise architecture effort.

(iii). **Business Organization**: The high-level nature of the organizational structures, business roles (internal audiences, external customers and partners), the decision-making process, and the organizational budget information..

2. **Enterprise Information View:** This is a view answers the question 'What capabilities are needed to support the business?' The Information view describes all of the moving pieces and parts for managing information across the enterprise, and the sharing of that information to the right people at the right time to realize the business objectives stated in the business architecture. The key components for describing the information architecture are:

   (i). **Information Strategy:** The information architecture principles, information governance and compliance requirements, canonical data models, and industry data model support strategy and a set of reference information exchange as well as dissemination patterns and reference models.

   (ii). **Information Assets:** A catalog of critical business data types and models (such as customer profile, purchase order, product data, supply chain, etc.) and the relationships between those business data types and all the services and processes that interact with that data.

   The Information view provides an information- and data-centric view of an organization, focusing on key information assets that are used to support critical business functions.

3. **Information System View:** The Information System view answers the question 'What applications and tools will be needed to support the business processes?' The information system view provides an application- and services-centric view of an organization that ties business functions and services to application processes and services to application components in alignment with the application strategy. The

information system's scope, strategy, standards are a consequence of the Business Architecture. The information system is composed of the following content categories:

(i). **Application Strategy:** The key application architecture principles (Build versus Buy, Hosted versus In-House, Open Standards versus .NET, etc.), application governance and portfolio management, and a set of reference application architectures relevant to the customer.

(ii). **Application Services**: An inventory of the key application services exposed to internal and external audiences that support the business services.

(iii). **Application Processes:** A series of application-specific processes that support the business processes in the Business Architecture.

(iv). **Logical Components:** An inventory of the relevant product-agnostic enterprise application systems that is relevant to the stated business objectives.

(v). **Physical Components:** The actual products that support the logical application components and their relationships to the relevant components and services in the information and technology architectures.

4. Infrastructure and Resource View: The ***Infrastructure and resource*** view answers the question 'What technology and resources (people and processes) will be used to enable the applications and tools?' adds knowledge about types and structure of components that support the information systems and actors. These may be hardware or network related. They may include fundamental services such as databases, etc. and key security and other commodity shared services. The infrastructure view provides a framework for specifying the technology elements of the organization's infrastructure. It includes: Platforms (hardware and software combinations supporting execution of applications), Networks, Data Storage and Management, Security, Internal Software Architecture of Applications, Middleware, User Interfaces, User/Function Interaction Models and Development Tools and Environments.

5. Governance and Security View: The ***Governance*** view answers the question 'What standards and procedures will guide the implementation of the architecture?' This view focuses on the manageability and quality of the architecture implementation (both business and IT) that is required to satisfy the services levels required by the business for its processes and systems. The artifacts for this area are all fundamentally defined within the core views (Business, Information, Information Systems and Technology Infrastructure and Resource), although the outcome from this view will be new specialized Services and Components to deliver the governance. The ***Security*** view focuses on the mitigation of known risks to the architecture implementation (both business and IT). The artifacts for this area are also all fundamentally defined within the core aspect areas (Business, Information, Information Systems and Technology Infrastructure and Resource). The outcome from this view will be new specialized Services and Components to deliver the required security.

3.3.5 Interrelationship of AIMES Views

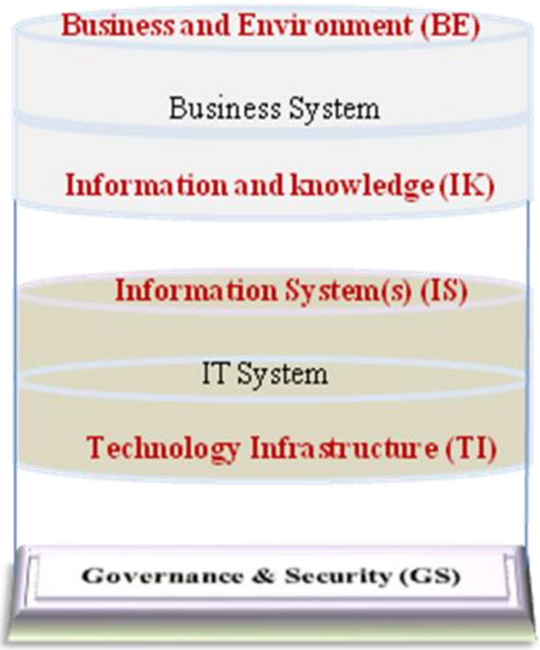


Figure 3.4 The views on the IT enabled enterprise.

The five main architecture views of AIMES are based on a “holistic” view on business and IT system of the IT enabled enterprise. In this view, the business is seen as two interrelated networks (Figure 3.4). The Business and Environment (BE) network consists of communicating and co-operating people in the role of employee, and of organizational units such as teams and departments. The network is organized as one or more supply chains of individuals, organizational units and companies working together in delivering products or services to the customers. The environment of a company is seen as network connecting the company with customers, suppliers and other third parties.

Information and knowledge is an important enabler of the business. The BE network is supported by an Information and Knowledge (IK) network formed by people and organizational units in specific IK supportive roles. These may be the same people and units that already have a role in the BE network.

The IK network enables the business by supporting the creation, exchange, storage and use of information and knowledge. The IK network in fact acts as the collective memory of the organisation.

The IT system that supports the business is also seen as a network system in two main layers: the information system(s) and the technology infrastructure. The information system(s) encompass a network of communicating and co-operating applications. The applications work together in delivering communication and information services to the people in the IT enabled Enterprise. These automated services enable the data processing, communication and control in the BE network, and the creation, exchange, storage and use of data in the IK network. The technology infrastructure is seen as a network of communicating and co-operating hardware devices and system software and middleware.

The Technology Infrastructure (TI) delivers processing, communication and storage capabilities to the information systems.

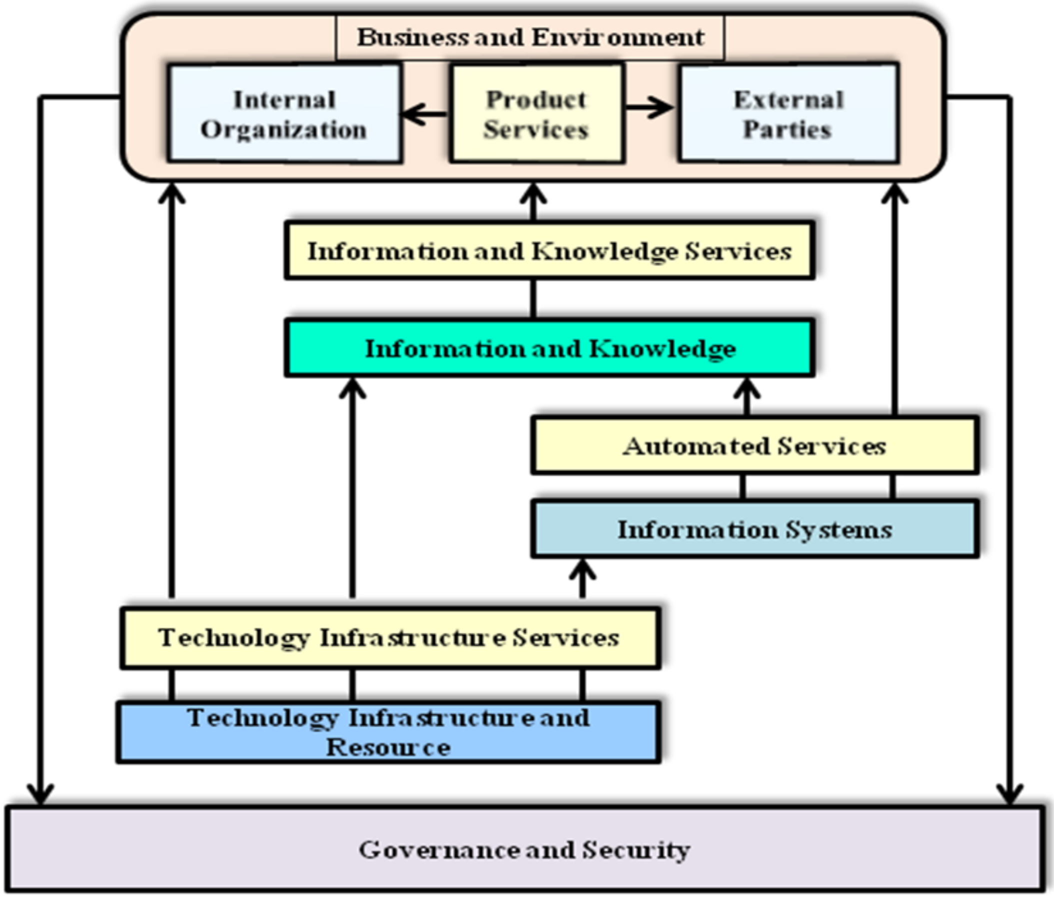


Figure.3.5 The enabling relationship between the architectural views

The main objective of AIMES is to support an architectural design of an IT enabled enterprise as one coherent co-operation of people, information, knowledge, applications and technology. The specific added value and benefits of AIMES are in the design and assessment of the enabling relationships (Figure 3.4), interactions, and dependencies among these architecture areas and not as much in the architectural design of the individual areas.

3.3.6 The AIMES Framework

Views / Phases		Enterprise Business Structure	Enterprise Information	Enterprise Information System	Technology Infrastructure & Resources	Governance & Security
Conceptual	**Strategic focus**	Business Strategy: Enterprise Definition, Mission, Vision, objective etc				
		Business System				
Building the Business Process and the Enterprise	**Architecture and Technology focus**	Business, Information, Information system, Infrastructure Architecture and Governance				
Logical – Design	**Delivery focus**	Information System				
		Business rule, logical data model, application architecture, Distributed system				
Implementation and Continuous Improvement		Hardware, Software, and Liveware (New business process requirements)				

Figure 3.6 The AIMES Framework

3.3.7 Merits of the Proposed Integrated Architecture Model for SMEs

i. Well-structured model, with strategic alignment of IT with the business and consistently defined information systems.

ii. Well-focused architecture design methodology that is simple, flexible and efficient.

iii. Well-structured architecture model for controlling enterprise IT systems complexity.

iv. Defines enterprise system as the instantiation of enterprise architecture in terms of a set of software modules, computer platforms, network components, and databases assembled in such a way as to be able to process business transactions and thus meet all the system requirements specified in that Enterprise Architecture.

v. Well-defined views used to classify applications and generated an information system framework for integrated enterprise system implementation.

3.3.8 Benefits of the Proposed Architecture

The proposed architecture will enable small and medium scale (SMEs) enterprises to;

i). Transform IT into an enabler for business by offering better alignment of business and IT;

ii) Deliver more flexibility for business and IT, whilst balancing the often contradictory needs of business.

iii) Manage IT System complexity better, mitigating risk and aiding overall decision-making.

3.3.9 Application of the AIMES Architecture

AIMES architecture has a strong orientation towards its practical application to enterprises. The model and method can be applied to any small and medium enterprises (SME's) as well as holdings and virtual enterprises of different sectors: chemistry, construction, manufacturing, transport, information technology etc. The objective is that it can be applied to both new enterprises (at the formation phase) and enterprises in the execution phase that want to improve their performance. However, at the moment all the work has been oriented more on real enterprises. For this reason, some of the tools that are going to be shown in this point should be adapted to the integrated design and implementation of new enterprises.

3.3.10 AIMES Modeling Approach – High Level Model

The high level model of this research is summarized in Figure 3.7. First, criteria are extracted from the characteristics of SMEs and EA approaches. Second, a first version of the AIMES metamodel will be developed, based on these criteria. Further, during case studies in SMEs, the metamodel will be refined and a method will be developed and refined. Third, the AIMES metamodel and method, in combination with criteria for developing integrated information system are being used to develop information system. This information system enables both the validation in the case study, as the validation by SMEs themselves that can use the system.

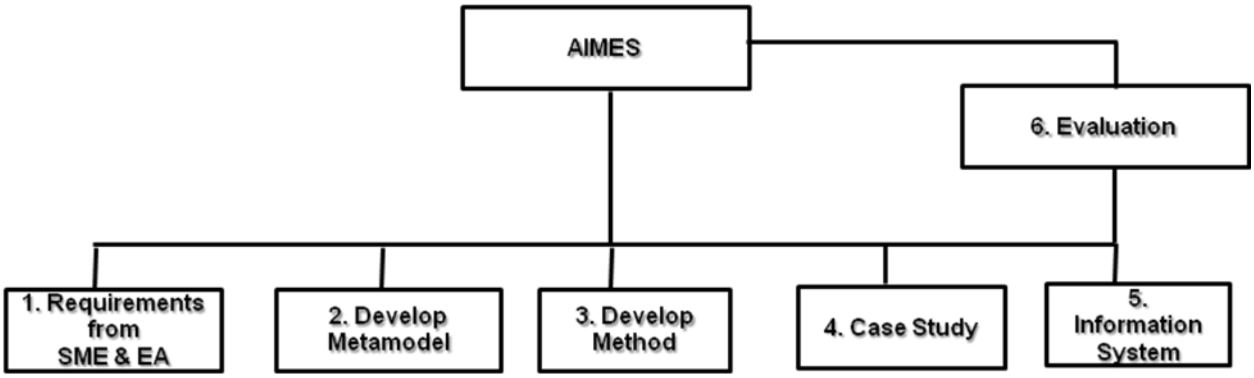


Figure 3.7 High Level Model for Developing an EA Approach for SMEs

CHAPTER FOUR
SYSTEM DESIGN

4.0 Introduction

System design is a step-by-step process of developing the structure of the model and the system. System design is an important phase of system engineering, determining system architecture (i.e. the way system components should be synthesized) to satisfy specific requirements. System design focuses on analyzing performance requirements, system modeling and prototyping, defining and optimizing system architecture and studying system design tradeoffs and risks. AIMES design is the process of defining and optimizing the architecture model (both enterprise and information system) and exploring performance requirements, ensuring that all the views are identified and properly allocated and that information system is well aligned with the SME vision in order to achieve a set of pre-defined goals.

4.1 Development Approach of AIMES for SMEs

In order to build an integrated enterprise in the SME using **AIMES,** the following requirements should be addressed:

i. Purpose of AIMES Architecture in the SME.
ii. Specific Business Questions for the SME.
iii. Assumptions and Business Rules for SME.
iv. AIMES Framework for SME
v. Metamodel of the SME - Donavan
vi. Models Needed in the Architecture for the SME
vii. Integrated representation of AIMES Metamodel (Integrated Metamodel) in the SME.

4.1.1 Purpose of AIMES Architecture in the SME

The primary purpose of the architecture is to optimize across the SME the fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy. This will develop and adapt EA approach specifically for SMEs in order to;

i. Standardize and streamline the business processes and become more efficient along with reducing overhead cost.
ii. Integrate the information systems and reduce the overhead of managing many disparate systems.
iii. Use better technology to improve process efficiency so the SME could be more competitive (Reduce technology costs)

4.1.2 Specific Business Questions for the SME

In order to achieve the purpose of the architecture for the SME, the proposed architecture will address the following questions:

i. What business activities and business processes are to be supported by the architecture in the SME?

ii. What are the system services, that is, the services provided by systems supporting business services in the SME?

iii. What applications support what business processes in the SME?

iv. What technology should be used for application and database? Will it be retired from the SME?

v. What applications use the data in the SME and how secure are they?

vi. What is the alignment between business processes, business activities, business information and support systems in the SME?

4.1.3 Assumptions and Business Rules for EA in SMEs

These are principles and guidelines intended to be enduring and seldom amended (The Open Group, 2009). They are ways to enhance the alignment within the enterprise.

According to IBM's (Schultz, 2007) and TOGAF's (The Open Group, 2009) conventions, the following fields were proposed for assumptions and business rule as shown in Table 4.1.

Table 4.1 Format for Assumptions and Business Rules

Field	Description
Name/Number	An easy meaningful name capturing the essence of the rule and a number for reference purpose.
Statement	Succinct and unambiguous fundamental rule
Rationale	Business benefit, intention of rule and relationship with other rules
Implication	Requirements for business and IT for carrying out the rule (eg. Resources, costs, activities)

The assumptions and business rules provide reasons/foundations for the use and deployment of IT resources and assets across the enterprise. They portray a level of consensus with the enterprise and provide a basis for future IT decisions. In other to fit AIMES, the following groupings are suggested.

i. Governance and management
ii. Business Architecture
iii. Information/data architecture
iv. Resource/infrastructure architecture
v. Solutions/information system architecture

Table 4.2 Assumptions and Business Rules for Process and Governance

Process, Governance and Management Rules	
PP1	**<u>Primacy of Rules</u>** These rules apply to all enterprises/units within the enterprise
PP2	**<u>Compliance with Law</u>** Enterprise information management processes comply with all relevant laws, policies and regulations
PP3	**<u>IT Responsibility</u>** The IT department/unit is responsible for owning and implementing IT processes and infrastructure which enable solutions to meet user-defined requirements for functionality, cost and delivery time
PP4	**<u>Protection of Intellectual Property</u>** The enterprise intellectual property must be protected and reflected in the IT infrastructure, implementation and governance processes.
PP5	**<u>On Demand</u>** An enterprise's business processes must be integrated end-to-end with partners, suppliers and customers. A business must rapidly respond to any customer demand, market opportunity or external threat.
PP6	**<u>Flexibility</u>** The architecture will incorporate flexibility to support changing business needs and to enable evolution of the architecture and the solution built on it.

PP7	**General Governance** Compliance to the architecture and evolution of the architecture will be managed through controlled governance processes.
PP8	**Cost Performance** The architecture will be managed to ensure the cost effectiveness of the information and technology environment.
PP9	**Application and Infrastructure Components** The application and infrastructure components will be designed and implemented in such a way as to facilitate monitoring and management.

Table 4.3 Assumptions and Business Rules for Business Architecture

Business Architecture Assumptions and Rules – BP	
BP1	**Maximize Benefit to the Organisation** Information management decisions are made to provide maximum benefit to the enterprise as a whole.
BP2	**Business Continuity** Enterprise operations are maintained in spite of system interruption.
BP3	**Common Use Application** Development of application used across the enterprise is preferred to the development of similar or duplicative application provided only to a particular section/unit.
BP4	**Service Orientation** The architecture is based on a design of services for real world business activities
BP5	**Technology Risk** Stability of business systems will be preserved through controlled usage and management of technology across its life cycle.
BP6	**Alignment of IT to Business** The architecture will be aligned with the business vision, objectives and strategies and will support the business operations
BP7	**Optimize IT infrastructure** The IT infrastructure will be optimized based on business requirements and technology capabilities.
BP8	**Strategic Use of Relations** The architecture will leverage strategic relationship with other businesses and vendor to facilitate the building and evaluation of the architecture.
BP9	**Enforce Security Policy** Implement processes, procedures and systems that promote enforcement of enterprise security policies.

Table 4.4 Assumptions and Business Rules for Information Architecture

Information/data Architecture (IP)	
IP1	**Data are an Asset** Data are an asset that has value to the enterprise and is managed accordingly
IP2	**Data are Shared** Data are shared across functions and units within the enterprise
IP3	**Data is Accessible**

	Data are accessible for users to perform their function.
IP4	**Data Trustee** Each data element has a trustee/owner accountable for data quality.
IP5	**Common Vocabulary and Data Definition** Data are defined consistently throughout the enterprise and the definitions are understandable and available for users.
IP6	**Data Security** Data are protected from unauthorized disclosure and use.
IP7	**Application Independence** Applications are independent of specific technology choices and can operate on a variety of technology platforms.
IP8	**Ease of Use** Applications are easy to use with the underlying technology transparent to users.
IP9	**Requirements-Based Access** A user (human or computer) should only be given enough privileges to do those tacks needed to perform specified job activity, functions.
IP10	**Information Confidentiality** All components of the computing environment must maintain confidentiality and integrity of the information that is used to conduct business, with decisions based on data classification

Table 4.5 Assumptions and Business Rules for Infrastructure (Technology) Architecture

Technology Architecture Assumptions and Rules – TP	
TP1	**Requirement-Based Change** Changes to applications and technology are only made in response to business needs.
TP2	**Responsive Change Management** Changes to the enterprise information environment are implemented in a timely manner.
TP3	**Control Technical Diversity** Technological diversity is controlled to minimize the cost of maintaining expertise in and connectivity between multiple processing environment.
TP4	**Interoperability** Software and hardware conforms to defined standard which promote interoperability of data, application and technology.
TP5	**Innovative and Agile** The architecture will readily support incorporation of new technologies to support business and technology innovation
TP6	**Technology and Vendor Independence** The architecture will be designed to reduce the impact of technology changes on the business as well as be resilient to change.
TP7	**Leverage Industry Knowledge** The architecture will leverage industry best practice.
TP8	**Open Standard** The architecture will use open standard.

Table 4.6 Assumptions and Business Rules for Information System (Solution) Architecture

Solution Architecture Assumptions and Business Rules –SP	
SP1	**IT Responsibility** The IT department/unit is responsible for owning and implementing IT processes and infrastructure which enable solution to meet user-defined requirements for functionality, service levels, cost and delivery timing.
SP2	**Single Point of View** Solutions provide a consistent, integrated view of the business regardless of access point.
SP3	**Buy versus Build** Business applications, system components and infrastructure will be purchased unless there is a competitive reason to develop them internally.
SP4	**Integrated solution** The architecture will support the delivery of business solutions composed of integrated application and infrastructure components.
SP5	**Completeness** A solution must be completely architected, including all viewpoints, solution portfolio and solution patterns.
SP6	**Reuse** Common components in the architecture should be used while balancing application and enterprise requirements. A solution reuses the as –built current state unless there are capabilities that need to be update.
SP7	**Real Solution** The described solution should be delivered and running within the enterprise and not just a piece of paper.
SP8	**Defence of Depth** Greater security will be obtained by layering defenses. Security control should be proportionate to risk.
SP9	**Security by Design** Security consideration should begin with requirement phase of development and be treated as an integral part of the overall system design
SP10	**Transparency** Security should be user transparent and not cause user undue extra effort.
SP11	**Limit Vulnerability** Design and operate IT systems to limit vulnerability and to be resilient in response.

4.1.4 Architecture Framework Used in the SME

The AIMES architecture framework was used. The AIMES architecture framework uses the *'business process'* as the structural *unit (*subsystem*)* on which the integrated enterprise system is developed. Within this framework, AIMES describes the architecture using two basic constructs. The first construct, ***Artifacts***, describes the architecture elements. Artifacts belong to, and are derived from specific areas in the architecture framework. The second, *Views*, is used to analyze and present the architecture from different perspectives and to document relationships between Artifacts. Views show why the architecture is what it is, providing both traceability of, and the justification for decisions made when developing the architecture. Views also provide a way to present architecture in an appropriate form for various stakeholders, thereby building acceptance as shown in Figure 3.6.

The AIMES architecture framework was used in conjunction with Archimate modeling language. This is because the core of Archimate contains the concepts and relationships that are necessary for general architecture modeling. Due to it being very visual and a rich language (i.e. many objects with distinct meaning), we have found the concept of a visual language a much easier way of describing the business, information, information system, resource and infrastructure and governance and security domain. It provides a vendor-independent, standardized set of concepts that helps to create a consistent, integrated model (Lankhost, 2009).

4.1.5 EA Metamodel of the SME – Donavan Tiles

The metamodel was designed based on the AIMES conceptual model domains as defined in figure 3.1.

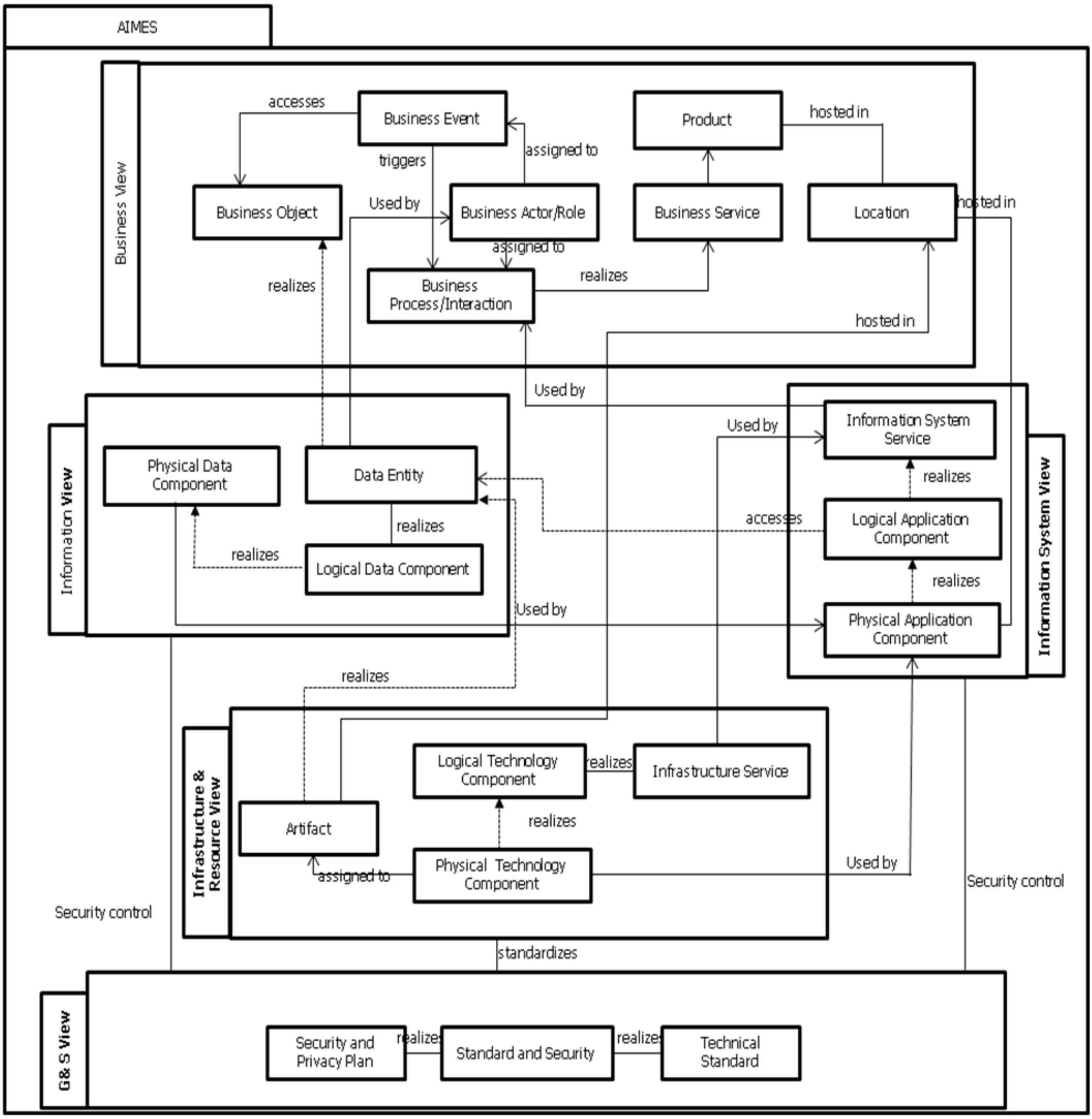


Figure 4.1 The EA Metamodel of the SME

The metamodel of the SME identifies a baseline set of physical entities, logical entities, and relationships among these entities. The metamodel of the SME shows the relationship between the architecture views (models). The *Business layer* is about business processes, services, functions and events of business units. This layer "offers products and services to external customers, which are realized in the organization by business processes performed by business actors and roles". The *Application layer* is about software applications that "support the components in the business with application services". The information layer is about data entities that ‘support the application components with data objects’. The *Technology layer* deals "with the hardware and communication infrastructure to support the Application Layer”. This layer offers infrastructural services needed to run applications, realized by computer and communication hardware and system software". The governance and security layer deals with general security of both the enterprise and information system and also the principles and procedures guiding the implementation of the EA in an SME.

4.1.6. Architecture Models Needed in the EA for SME

In other to create the integrated model for the SME and answer the business questions, we distinguished five main layer models and three alignment models:

i. The Business layer model offers products and services to external customers, which are realised in the organization by business processes (performed by business actors or roles).

ii. The Information system layer model supports the business layer with application services which are realised by (software) application components.

iii. The Information layer model supports the information system layer by providing the data objects that work on a subject and conveys meaningful information which are realised by artifacts

iv. The Infrastructure and Resources layer model offers infrastructural services (e.g., processing, storage and communication services) needed to run applications, realised by computer and communication devices and system software.

v. The Governance and security model offers general architecture governance and provides relevant technology standards needed for the architecture management.

v. The Business layer – Information system alignment model.

vi. The Information system – data element alignment model.

vii. The information system, data element and infrastructure alignment model.

4.1.6.1 The AIMES Business View

The *Business* view adds knowledge about business objectives, activities, and organizational structure. The Business component deals with the enterprise and its environment, as well as the architectural elements within it. The types in this section are designed to support the modeling of an enterprise, the context within which it operates, and its interactions with parties in this environment, including customers, partners, shareholders etc.

The types cover issues such as: Customers and Markets, Products and Services, Channels, Organizational Units, Deals (contractual or financial arrangements between parties), Business Processes, Culture, Competitors, Business Goals and various other aspects.

Business view focuses on the purpose, scope and policies of the system and provides the overall environment in which the system will be built. It will capture business processes automated by the system to give it a holistic, process-centric view. This view possesses the processes, the participants of the process: applications and people, the roles played by various users, the location where the application would be deployed etc. this view describes various functional blocks (or subsystems) of the system, the functional services offered by them, their inter-relationships, the information exchanged by them and so on. It focuses on the formalization of the construction process from the model of the different enterprise structures.

Based on the notion of business system, we built a business view metamodel that identifies and represents the set of generic concepts that may be found in any business system. This meta-model is shown in Figure 4.1. The importance of this model is that it identifies the set of business concepts that must be represented during the process of modeling the application domain of an Enterprise Information System - 'The Total Information Processing System'. A modelling method like the enterprise Organigram can be used. Table 4.1 shows the business view concepts and there description. The Key artifacts in this view are as shown by the meta-model in Figure 4.1:

Table 4.7 The Business View Concepts

Concept	**Description**
Business Actor	An organisational entity that is capable of performing behaviour
Business Role	The responsibility for performing behaviour to which an actor can be assigned
Business Collaboration	An aggregate of two or more business roles that work together to perform a behaviour
Business Interface	A point of access where a business service is made available to the environment
Location	A conceptual point or extent in space
Business Object	A passive element that has relevance from business perspective
Business Process	A behaviour element that groups behaviour based on an ordering of activities. It is intended to produce a defined set of product or business services
Business Function	A behaviour element that groups behaviour based on a chosen set of criteria (typically required business resources and/or competencies)
Business Interaction	A behaviour element that describes the behaviour of a business collaboration
Business Event	Something that happens (Internally or externally) and influences behaviour
Business Service	A service fulfils a business need for a customer (internal or external to the organisation).
Representation	A perceptible form of information carried by a business object
Meaning	The knowledge or expertise present in a business object or its representation given a particular context
Value	The relative worth, utility or importance of a business service or product
Product	A coherent collection of services, accomplished by a contract/ set of agreements which is offered as a whole to (internal or external) customers
Contract	A formal or informal specification of agreement that specifies the rights and obligations associated with a product

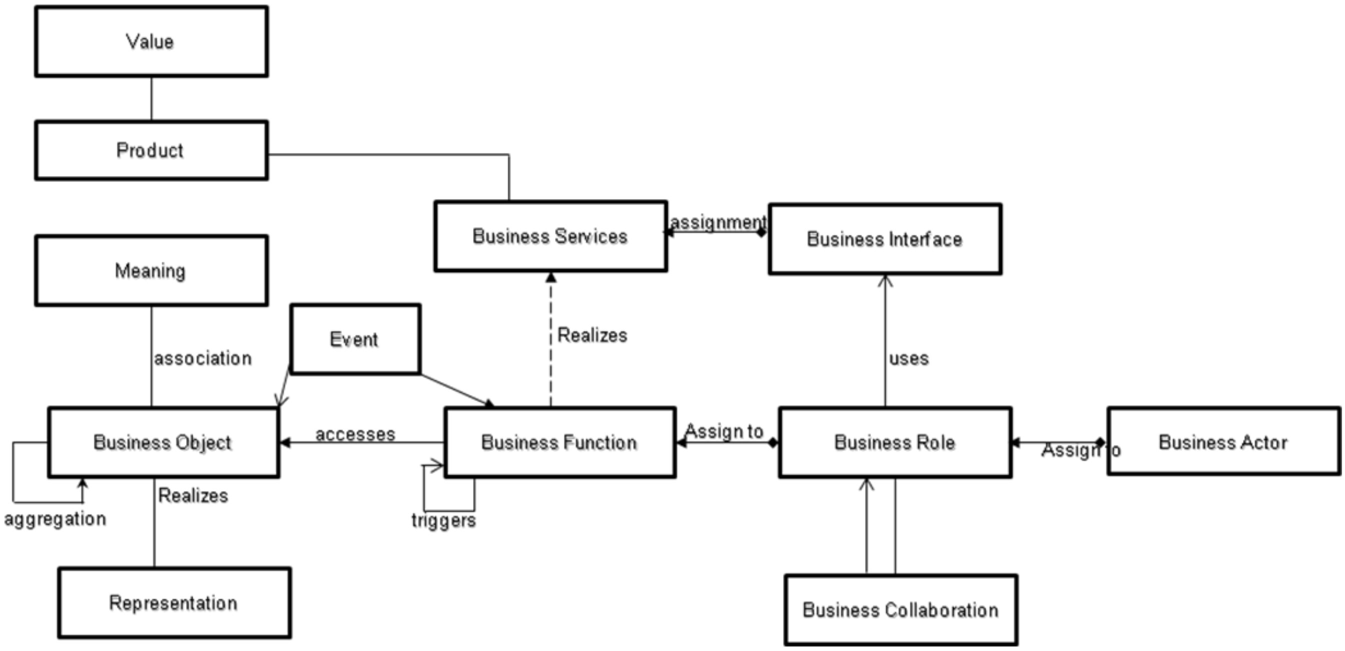


Figure 4.2 AIMES Business View Metamodel (Jonkers et al., 2001) Modified by Author

Relationship with other Views

The Business layer concepts are related to the concepts in information and information system view as follows;

i. An activity/role can be mapped to supporting applications from the information system view.
ii. An information exchange is represented in the information view.
iii. A data object can be described by one or more information view classes or entities

4.1.6.1.1 Application to the SME: Donavan Business Layer Model

The main structural concepts at the business layer are business roles and business actors, an entity that performs behaviour such as business processes or functions. A *business role* signifies responsibility for one or more *business processes* or *business functions*. A business function denotes the high-level capabilities of an organization, and offers functionality that may be used in business processes to realize the products and services of the organization.

Business functions can be connected through *flows* that describe the information or goods exchanged. A business role is typically assigned to a business actor. *Business actors* may be individual persons (e.g. customers or employees), but also groups of people and resources that have a permanent (or at least long-term) status within the organizations. Business processes, which may be triggered by *events* and manipulate *business objects*, describe the business behaviour of a role. The externally visible behaviour of a business process is modeled by the concept of *business service*, which represents a unit of functionality that is meaningful from the point of view of the environment. Not shown in the example is that services can be grouped to form (financial or information) *products*, together with a *contract* that specifies the associated characteristics, rights and requirements.

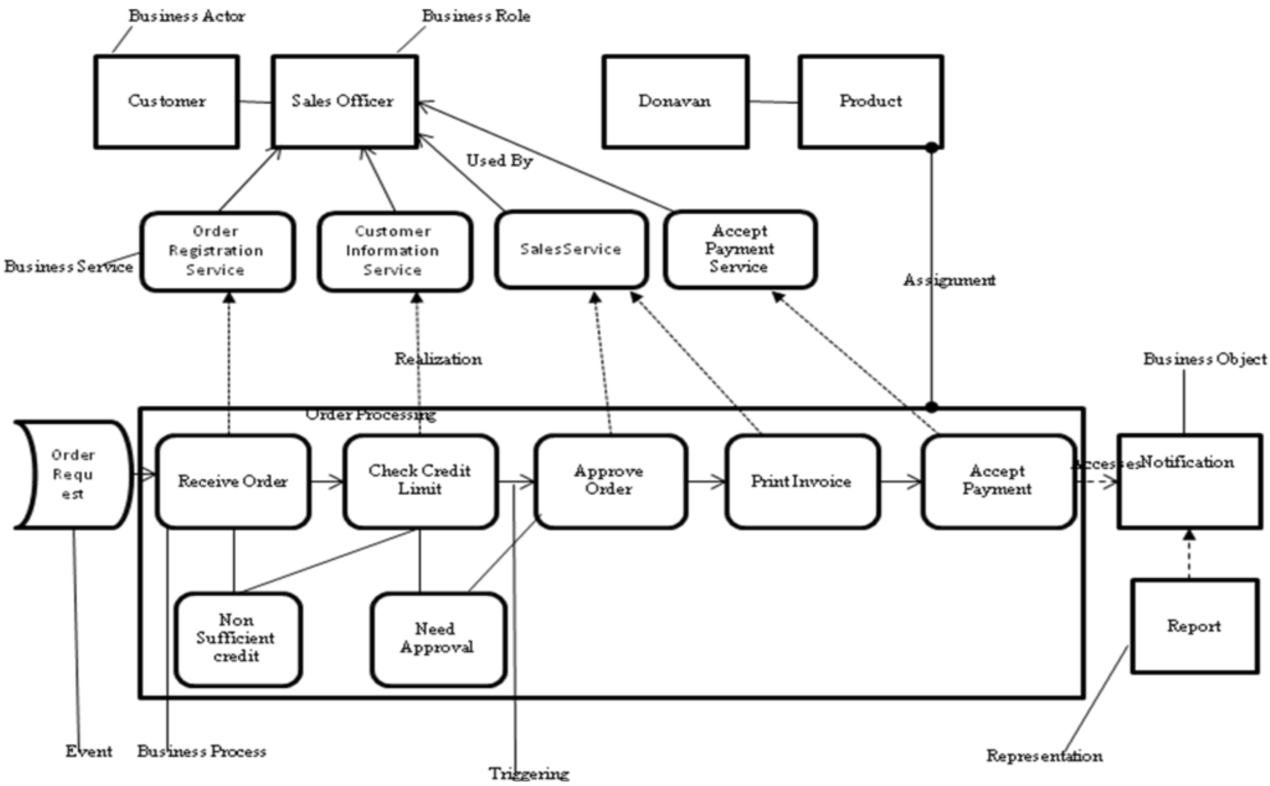


Figure 4.3 Sample Donavan Business layer model

4.1.6.2. The AIMES Information View

The *Information* view adds knowledge about information that the business uses, its structure and relationships. Information is fundamental to the successful operation of a business and we need accurate and timely information. Information view describes what the organisation needs to know to run its processes and operations as described in the business architecture. It defines a view on the business information that is system and technology independent. It is the abstraction of the information requirements of the enterprise and provides a high level representation of all the key information elements that are used in the business as well as the relationship between them. Business information is structured as a collection of information entities. An entity can result from the composition or specialization of other entities in the object oriented sense. Information entities are classes meaning they can be typified. Entities describe most artifacts of the enterprise namely those resources required by processes, including business, support and management processes. As such, they have identifier, defined from business perspective and a set of attributes. Attributes are related to roles the entities play. Therefore, each role integrates its set of attributes into the entity. The overall set of attributes results from merging each individual set of attributes derived from each role the entity is able to play. Table 4.8 shows the Information view concepts. Key artifacts in this view are as shown by the meta-model in Figure 4.8:

Table 4.8 Information View Concepts

Concept	Description
Data function	A behaviour element that groups data/information based on a chosen criteria
Data Interface	A point of access where data/information service is made available to IS component or user
Information Service	A Service that exposes information behaviour
Physical data Component	An entity that contains artifacts which have identifier and attributes
Data entity/element	A data subject or group of data that work on a subject and conveys meaningful information

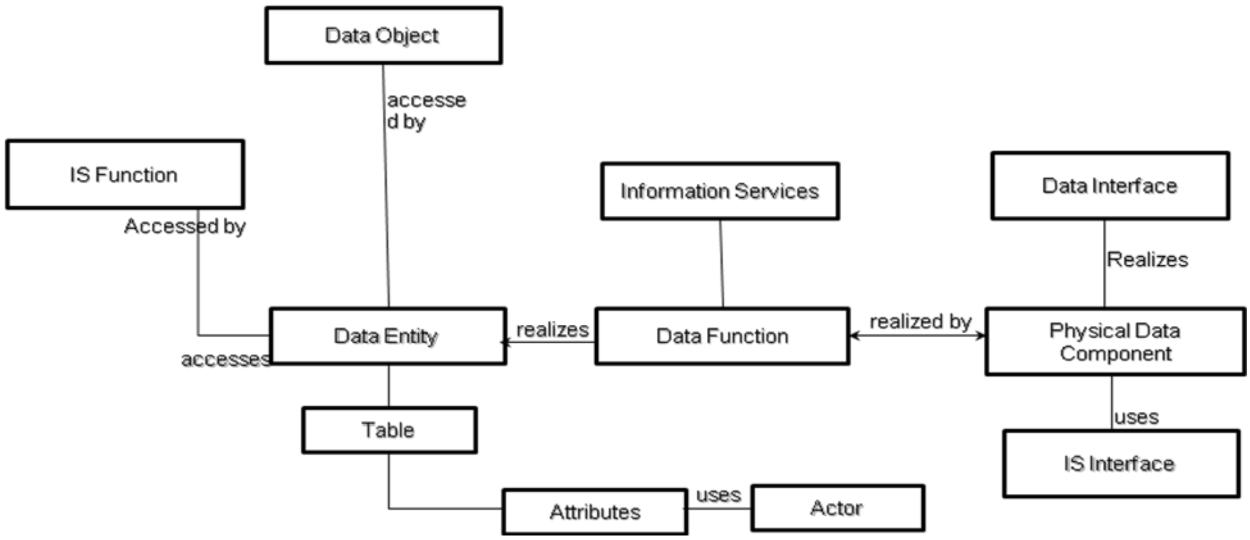


Figure 4.4 AIMES Information View Metamodel (by author)

The information view is a model that contains the individual data models developed for each business area and also defines the interrelationship between those models. This view also contains the business rules based on the process analysis and on the interaction of process and data. Figure 4.4 is an overview of the AIMES information view which focuses on supporting the following business requirements:

i. Data organization and management to facilitate broad access.
ii. Intelligent tools and systems for user and information profiling
iii. Data standardization
iv. Data dictionary for storage and retrieval of selected corporate information
v. Information security and integrity

Relationship with other Views

The Information view concepts are related to business view concepts and information system view concepts in the following ways;

i. Entities can be contained with data objects defined in the business view
ii. Entities can be included in or mapped to the enterprise data models based on FEA data reference model.
iii. Tables can be managed by information systems

4.1.6.2.1 Application to the SME: Donavan Enterprise Information Layer Model

Data objects are used in the same way as data objects (or object types) in well-known data modelling approaches, most notably the 'class' concept in UML class diagrams. The data object describes most artifacts of the enterprise, namely those resources required by processes, including business, support and management processes. As such, they have an identifier, defined from a business perspective, and a set of attributes.

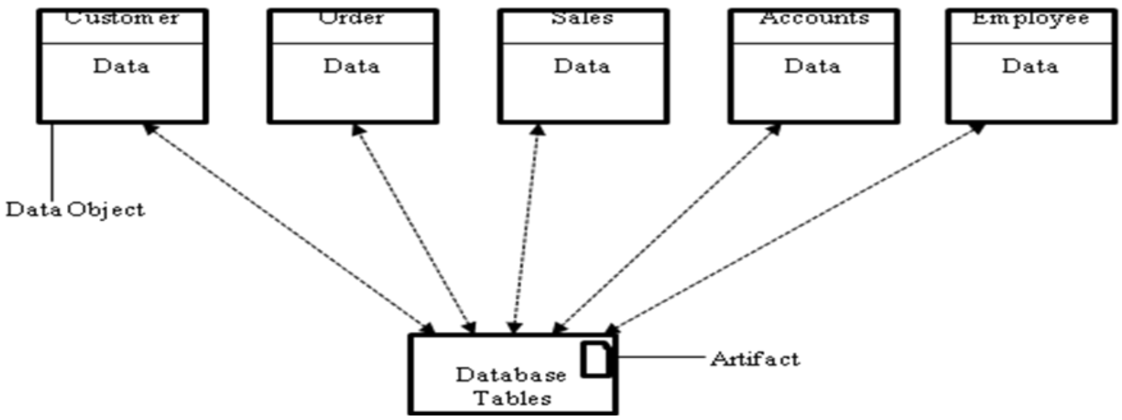


Figure 4.5 Sample Donavan Enterprise information model

4.1.6.3. The AIMES Information System View

The *Information System* view adds knowledge about types of information systems (packaged or bespoke) that can automate and support the processing of information used by the business. The Information System view fulfils two goals (i) supporting the business requirements and (ii) allowing efficient management of the enterprise's entities. To satisfy these goals, the Information system view should be derived top-down from the analysis of the business and information views. The information system view defines the applications needed for data management and business support regardless of the actual software used to implement systems. It functionally defines what application services are required to ensure processes and entities are supported in acceptable time, format and cost. Figure 4.6 shows the meta-model of the enterprise information system.

Key artifacts include:

- Information System Service
- Logical Information System Component
- Physical Information System Component

Table 4.9 Information System View Concepts

Information System Component	A modular, deployable and replaceable part of a software system that encapsulates its behaviour and data and exposes these through a set of interface
Information System Collaboration	An aggregate of two or more IS components that work together to perform collective behaviour
Information System Interface	A point of access where an IS Service is made available to a user or another application component
Data Object	A passive element suitable for automated processing
Information system function	A behaviour element that groups automated behaviour that can be performed by an IS component
Interaction	A behaviour element that describes the behaviour of an IS collaboration
Information System Services	A service that exposes automated behaviour.

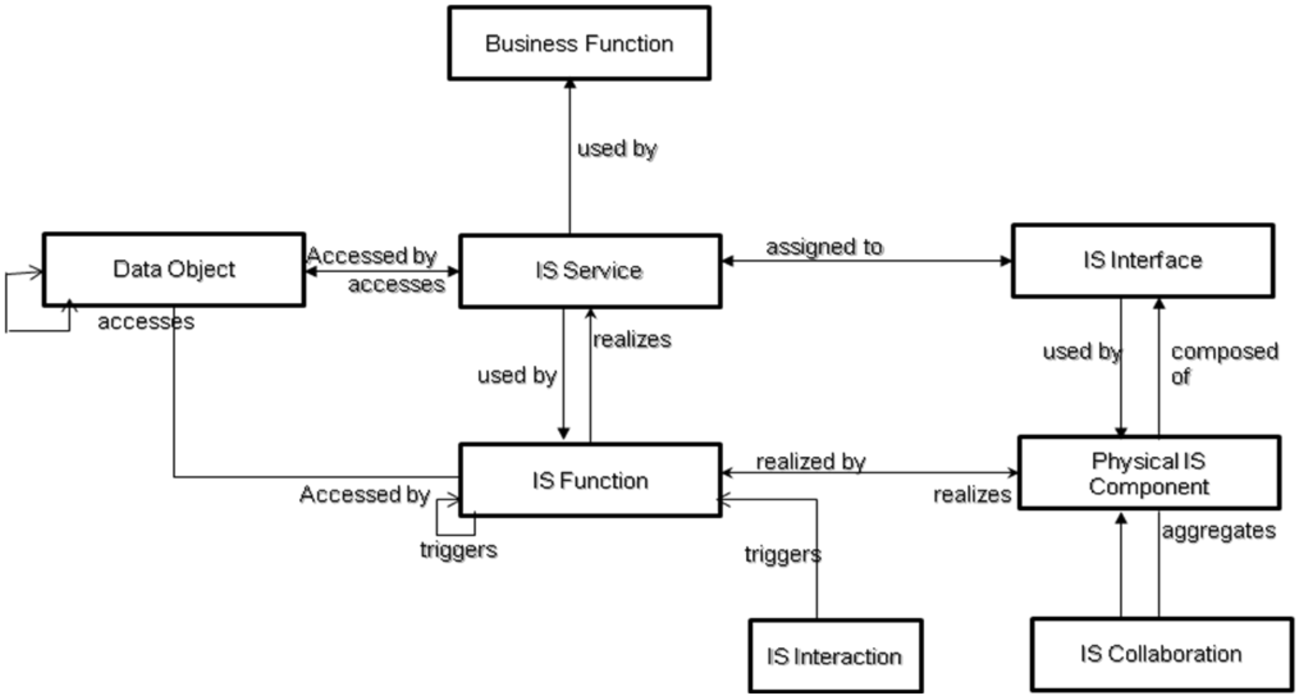


Figure 4.6 AIMES Information System View Metamodel (Jonkers et al., 2001) Modified by Author

Information systems are computer systems applied to the solution of business problems or realization of organizational or personal goals. They often directly interact with and support the work of end users. Typical examples would include a finance system, debtors system; order processing, bank account management, personnel etc. Information system types are likely to change over time and the mix can dramatically change the required infrastructure.

Relationship with other Views

Concepts in the information system are related in other views as follows;

i. Information system can be used by activities in the business view.
ii. Information system can manage data in tables defined in the information view.
iii. Information can be hosted on platforms in the infrastructure and resource view.
iv. Solution components can be dependent upon or built using technologies identified in the governance and security view

4.1.6.3.1 Application to the SME: Donavan Information System Model

The main structural concept for the application layer (Figure 4.7) is the *application component*. This concept can be used to model any structural entity in the application layer: not just (reusable) software components that can be part of one or more applications, but also complete software applications or information systems. Behaviour in the application layer can be described in a way that is very similar to business layer behaviour. Again, there is a distinction between the externally visible behaviour of application components in terms of *application services*, and the internal behaviour, *application functions*, that realize these services1. Services are offered through the *application interfaces* of an application.

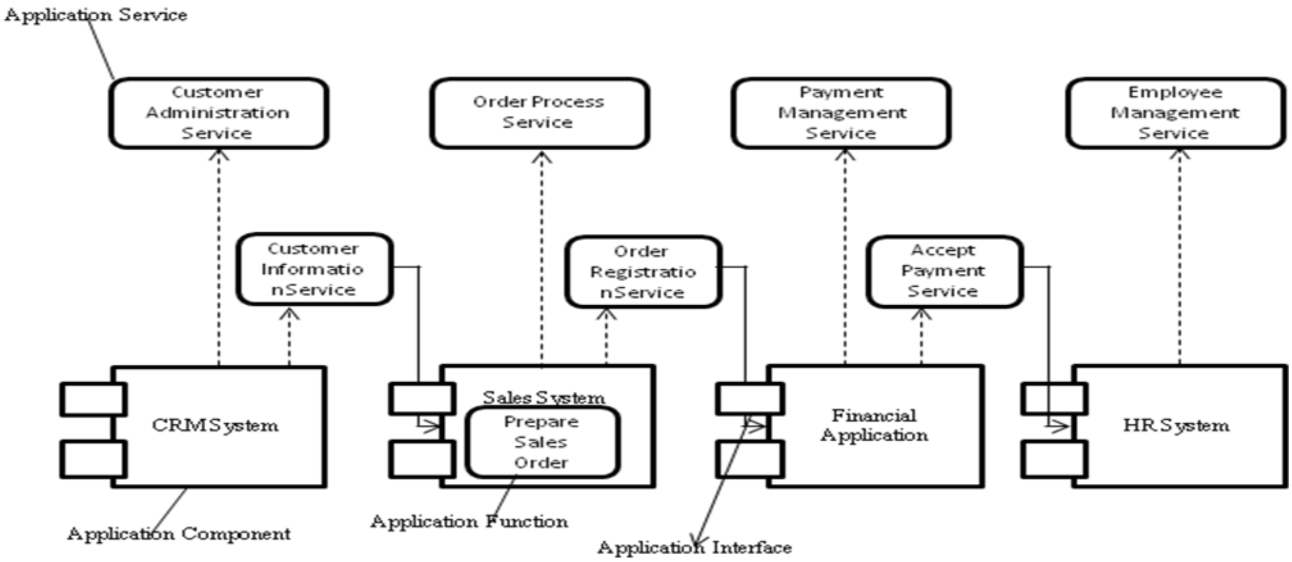


Figure 4.7 Sample Donavan Information System model

4.1.6.4. The AIMES Infrastructure and Resource View

The *Infrastructure and resource* view adds knowledge about types and structure of components that support the information systems and actors. These may be hardware or network related. They may include fundamental services such as databases, etc. and key security and other commodity shared services.

The infrastructure view provides a framework for specifying the technology elements of the organization's infrastructure. It includes; Platforms (hardware and software combinations supporting execution of applications), Networks, Data Storage and Management, Security, Internal Software Architecture of Applications, Middleware, User Interfaces, User/Function Interaction Models and Development Tools and Environments.

This view is normally described at the level of types of things, rather than instances. This is to avoid the problem of excessive detail bogging down the architecture process and strategy. It is possible to track the infrastructure elements at an instance level if desired, but this requires considerable commitment, effort and dedicated resources.

Being a resource is not an innate property of an object but is derived from the role an object plays with respect to an activity. Primitive resource properties are identified and defined (Figure 4.8), from which more complex properties would be defined. An activity uses a resource. A resource has *quantity, location, and role* and is governed by *restriction*. Statuses of a resource are *production, consumption, and use*. A resource can be divided into *continuous and discrete*. A continuous resource is uncountable, whereas discrete resources such as finance, human, and supplier are countable. The infrastructure and resource view meta-model is as shown in Figure 4.8.

Key artifacts include:

- Technology Infrastructure Service
- Logical Technology Infrastructure Component
- Physical Technology Infrastructure Component
- Users and Employees

Table 4.10 Infrastructure and Resource View Concepts

Concept	Description
Node	A computational resource upon which artifacts may be stored or deployed for execution
Device	A hardware resource upon which artifacts may be stored or deployed for execution
Network	A communication medium between two or more devices
Communication Path	A link between two or more nodes through which these nodes can exchange data
Infrastructure Interface	A point of access where infrastructure services offered by a node can be accesses by other nodes and information system components
System Software	A software environment for a specific type of component and object that are deployed on it in the form artifacts
Infrastructure function	A behaviour element that groups infrastructural behaviour that can be performed by a node
Infrastructure Service	An externally visible unit of functionality, provided by one or more nodes exposed through well defined interfaces and meaningful to the environment
Artifact	A physical piece of data that is used or produced in a software development process or by deployment and operation of a system

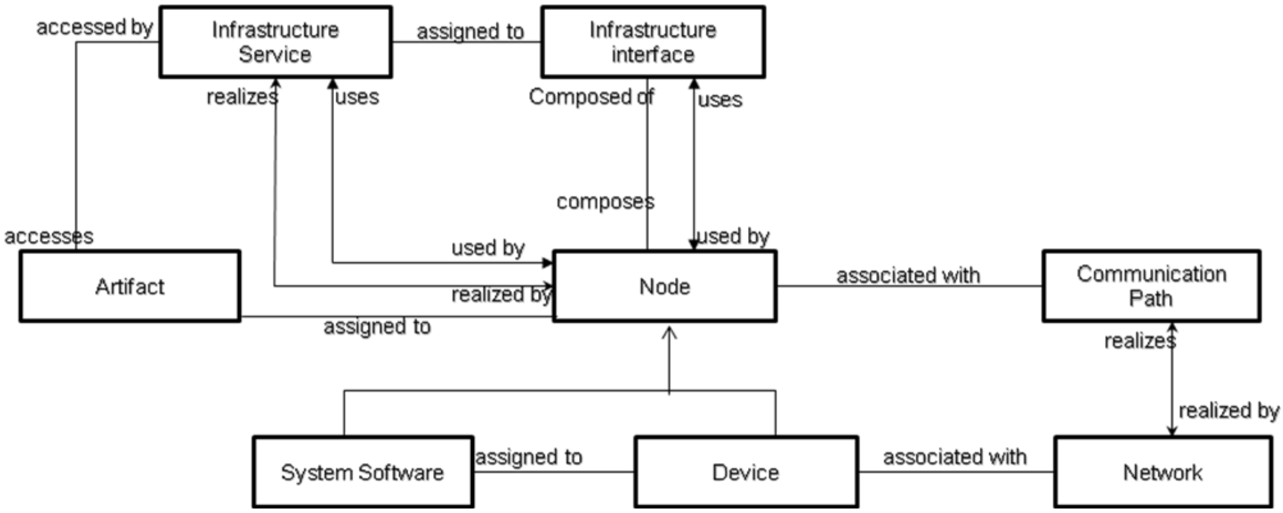


Figure 4.8 AIMES Infrastructure and Resource View Metamodel (Jonkers et al., 2001) Modified by author

4.1.6.4.1 Application to the SME: Donavan Infrastructure and Resource Layer Model

The main structural concept for the technology layer is the *node*. This concept is used to model structural entities in the technology layer. Nodes come in two forms: *device* and *system software*. A device models a physical computational resource; system software represents the software environment for specific types of components and data objects. An *infrastructure interface* (not shown in the example) is the (logical) location where the infrastructural services offered by a node can be accessed by other nodes or by application components from the application layer.

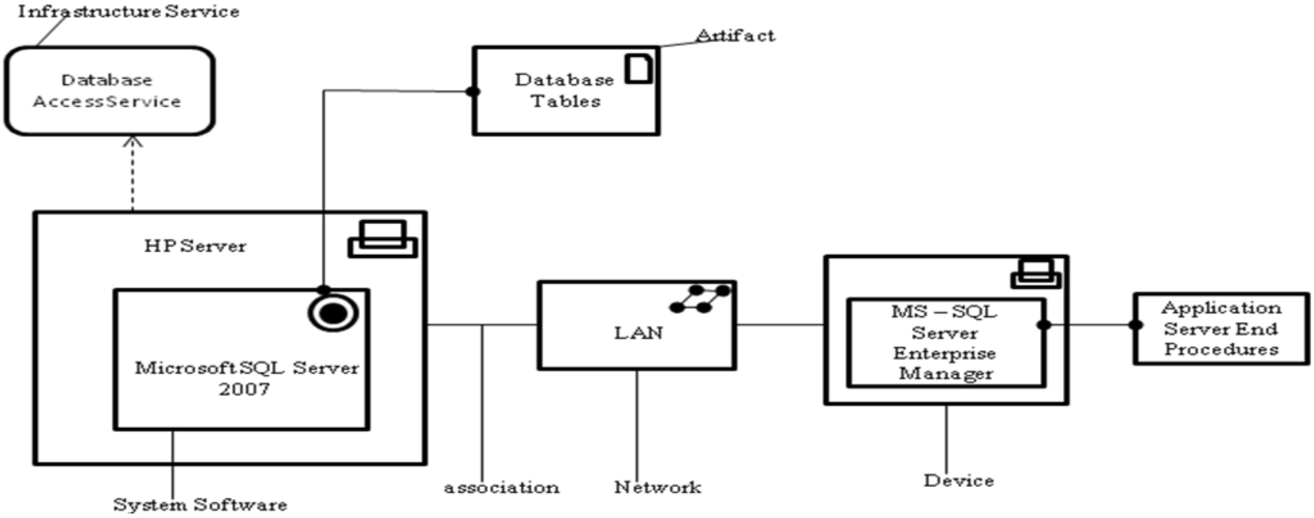


Figure 4.9 Sample Donavan Infrastructure and Resource model

4.1.6.5 AIMES Governance and Security View

The *Governance and Security* view focuses on the manageability, security and quality of the architecture implementation (both business and IT) that is required to satisfy the services levels required by the business for its processes and systems. The artifacts for this area are all fundamentally defined within the core views (Business, Information, Information Systems and Technology Infrastructure and Resource), although the outcome from this view will be new specialized Services and Components to deliver the governance. The view also focuses on the mitigation of known risks to the architecture implementation (both business and IT). The artifacts for this area are also all fundamentally defined within the core aspect areas (Business, Information, Information Systems and Technology Infrastructure and Resource). The outcome from this view will be new specialized Services and Components to deliver the required management and security.

Key artifacts include:

- Service Delivery
- Security services
- Policy and standard
- Control

Table 4.11 Security and Governance View Concepts

Concept	
Policy	A basic principle and guideline for service assessment
Standard	A basic principle used to measure quality and performance
Standard Platform	A standard hardware and software to be used for deployment and execution
Service Quality	A behavioral assessment of the services offered by IT and Business
Metrics	A standard for measuring efficiency and quality
Control	A behaviour that influences standard and performance

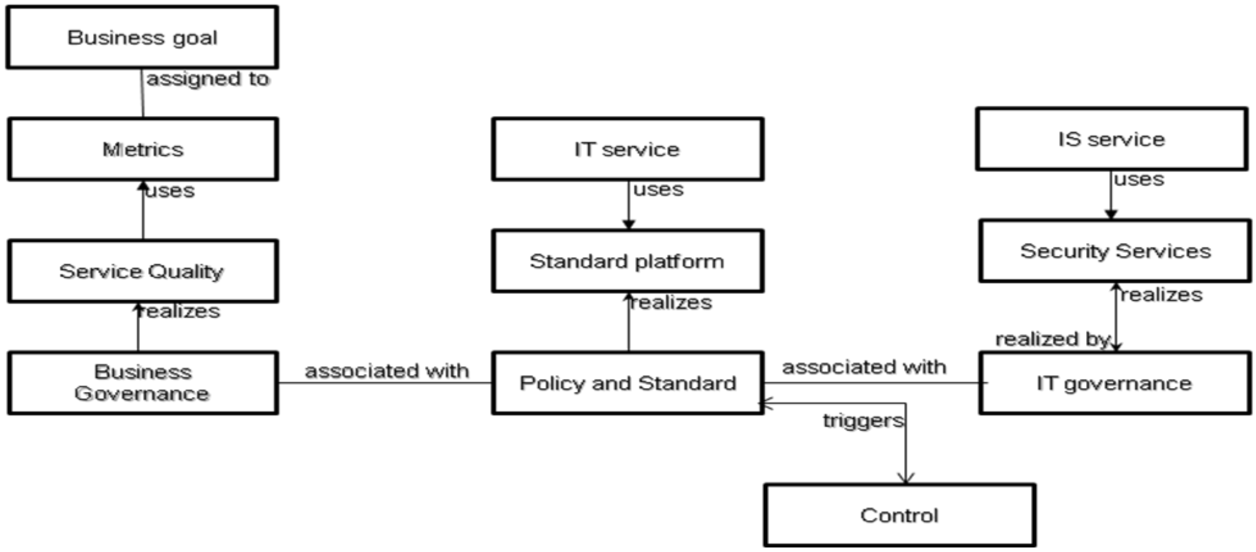


Figure 4.10 AIMES Governance and Security View Metamodel (by Author)

4.1.6.5.1 Application to the SME: Donavan Governance and Security View

The governance and security view includes security and privacy plans and identifies relevant technology standards and their forecasted evolution. Governance is required for decision making. Governance helps define the rules and strategies needed in the architecture.

4.1.6.6. Business– Application Layer Alignment Model

The application layer and the business layer can be easily linked. One type of relation provides this link. This shows cross-layer dependencies between business layer and application layer.

(i). Application services can be *used by* business behaviour and application interfaces are *used by* business actors roles, i.e., there is a *support* relation between the application and business layers.

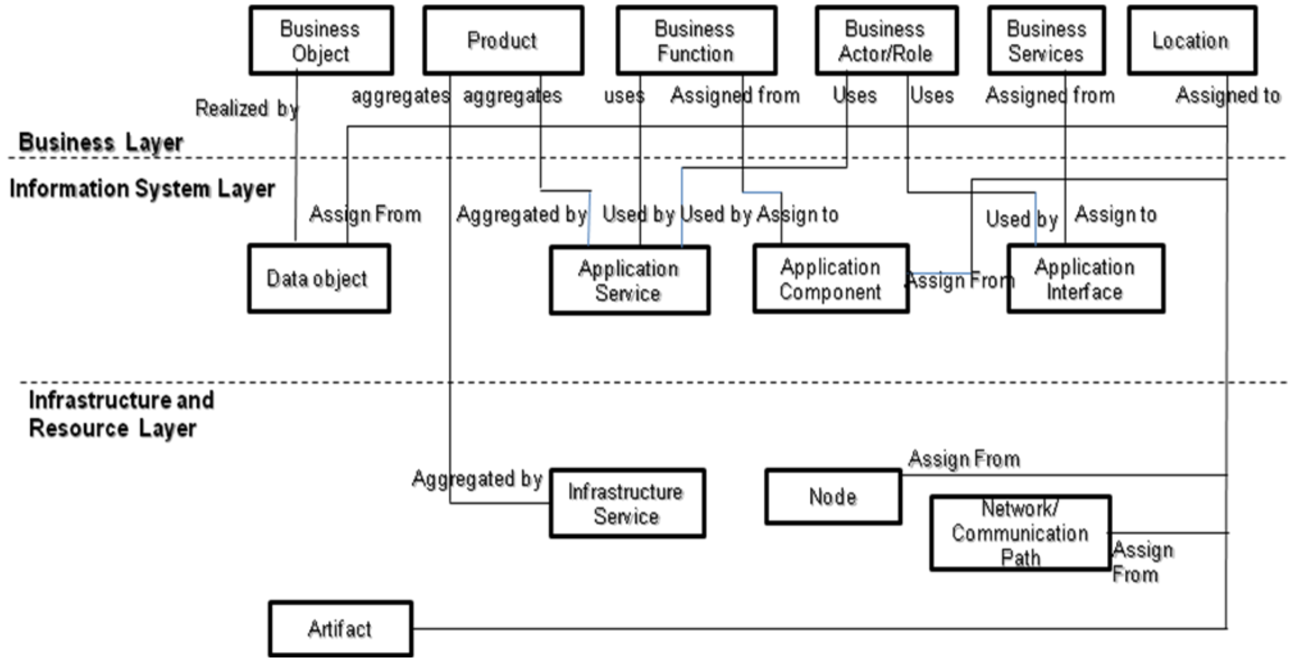


Figure 4.11 Business – Application Layer Alignment Model

4.1.6.6.1 Application to the SME: Donavan Business Process, Services and Application Alignment Model

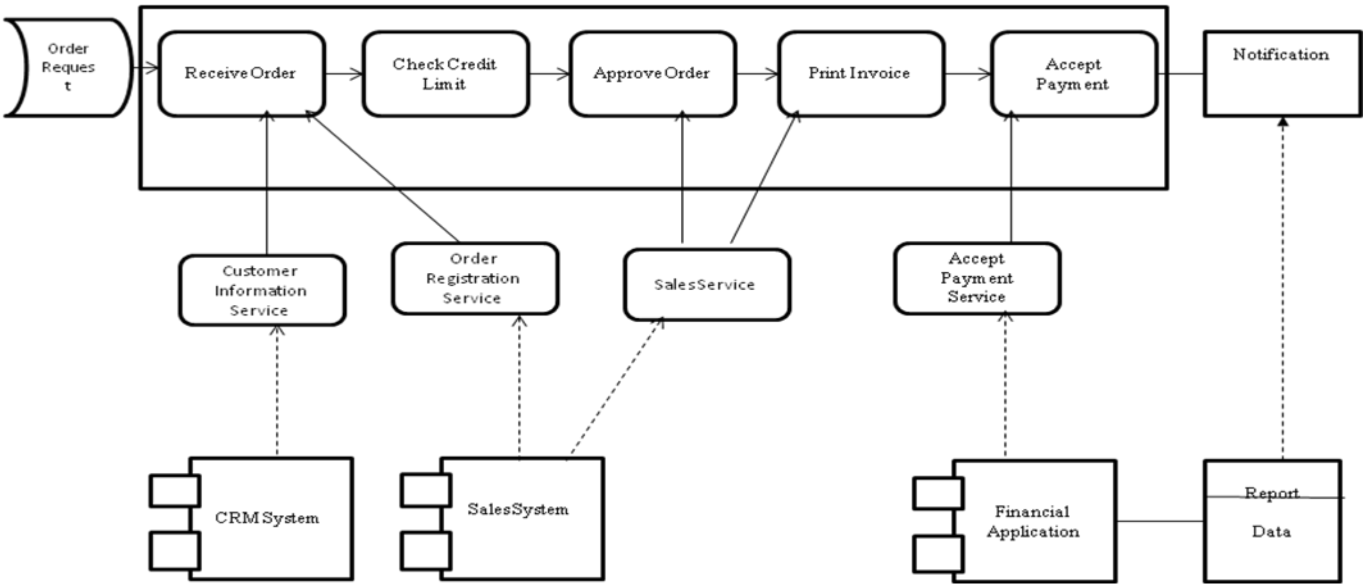


Figure 4.12 Business Process – Application Alignment Model

Misalignments often drive people to engage in other tasks than those required by the activities being performed. The need for extra tasks is an evidence of misalignment and is measured in extra time required by business people to conduct and fulfill the business.

4.1.6.7. Information System – Information Alignment Model

The information layer and the Information system layer can also be linked very easily. Similar to business-application alignment, one type of relation provide this link:

(i) Data entities can *realize* data objects; this means that a data object is an electronic *representation* of the business object, i.e., there is an *implementation* relation between the application and information layers.

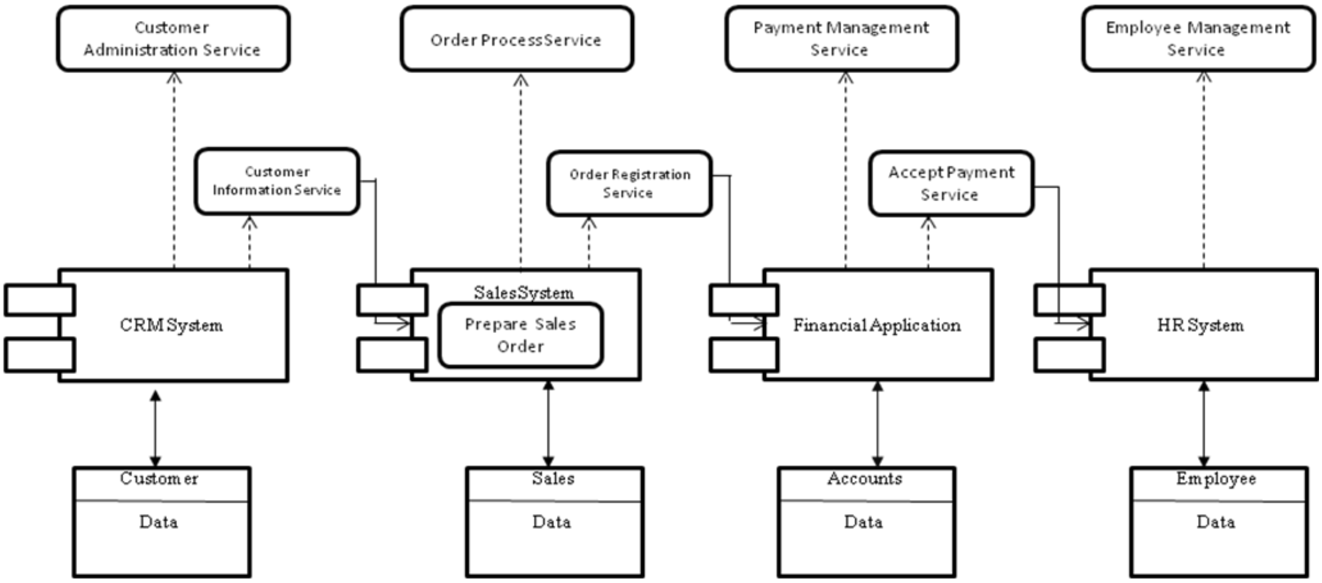


Figure 4.13 Information Systems – Information (Data element) Alignment Model

Misalignments between information and application architectures result in extra time of IT department in ensuring that applications have the right data for processing. This means that IT people either spend time in keeping information entity replicas coherent, or spend time in integration projects that serve no other purpose than assuring information replicas coherency. In both cases, the extra time and money are an evidence of misalignment between information and application architectures.

4.1.6.8. Information System - Infrastructure Alignment Model

The technology layer, information layer and the application layer can also be linked very easily. Similar to business-application alignment, two types of relations provide this link:

(i) Infrastructure services can be *used by* application functions and infrastructure interfaces are *used by* application components, i.e., there is a support relation between the technology and application layers (Figure 4.14).

(ii) Artifacts can *realize* data entities, data objects and application components, that is, there is an implementation relation between the technology and information layers (Figure 4.14).

Artifacts play a central role in showing how 'logical' application components are realised by 'physical' components (modeled as artifacts). A single physical component may realize multiple logical components and, conversely, multiple physical components may be used to realize a single logical component.

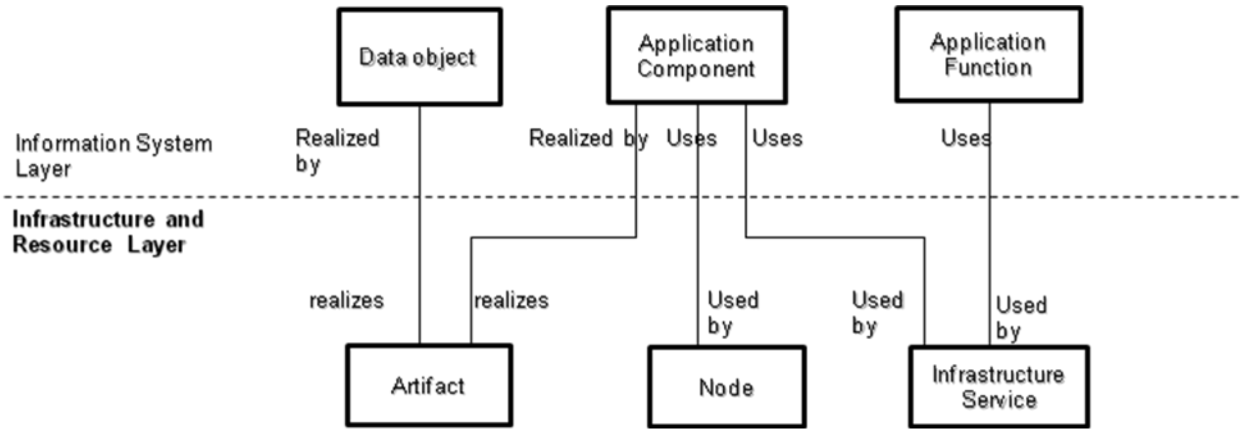


Figure 4.14 Information System – Infrastructure Alignment Model

4.1.6.8.1 Application to the SME: Donavan Information system - Infrastructure Alignment Model

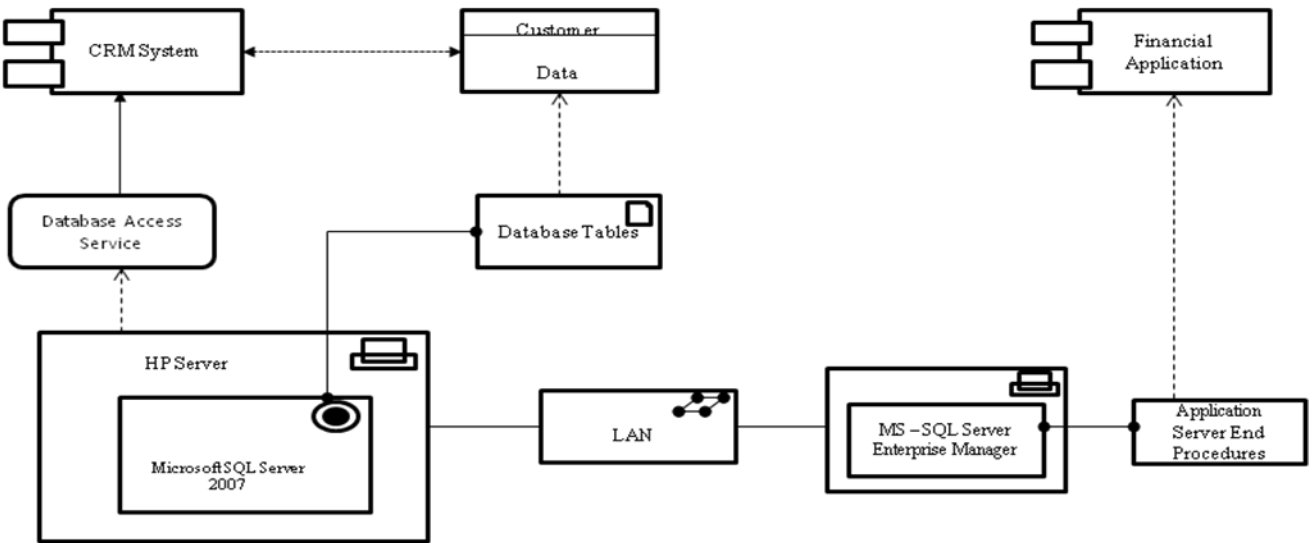


Figure 4.15 Sample Donavan Infrastructure, Information and Information System Alignment Model

4.1.7 Integration of the Architecture

The integrated metamodel shows cross-view dependencies with the service concept as the linking pin. Each view makes their external service available to the next view. The external services of the higher view may depend on the service in the same architectural view or one view below as shown in figure 4.16. The figure shows an overview of an integrated and service-oriented enterprise architecture model. This was constructed by connecting models from different layers by means of services and realization relations. The metamodel shows the relationship between the architecture models which include the business model, information system model, information model and infrastructure and resource model. This is based on a layered architecture metamodel concept as shown in figure 4.16.

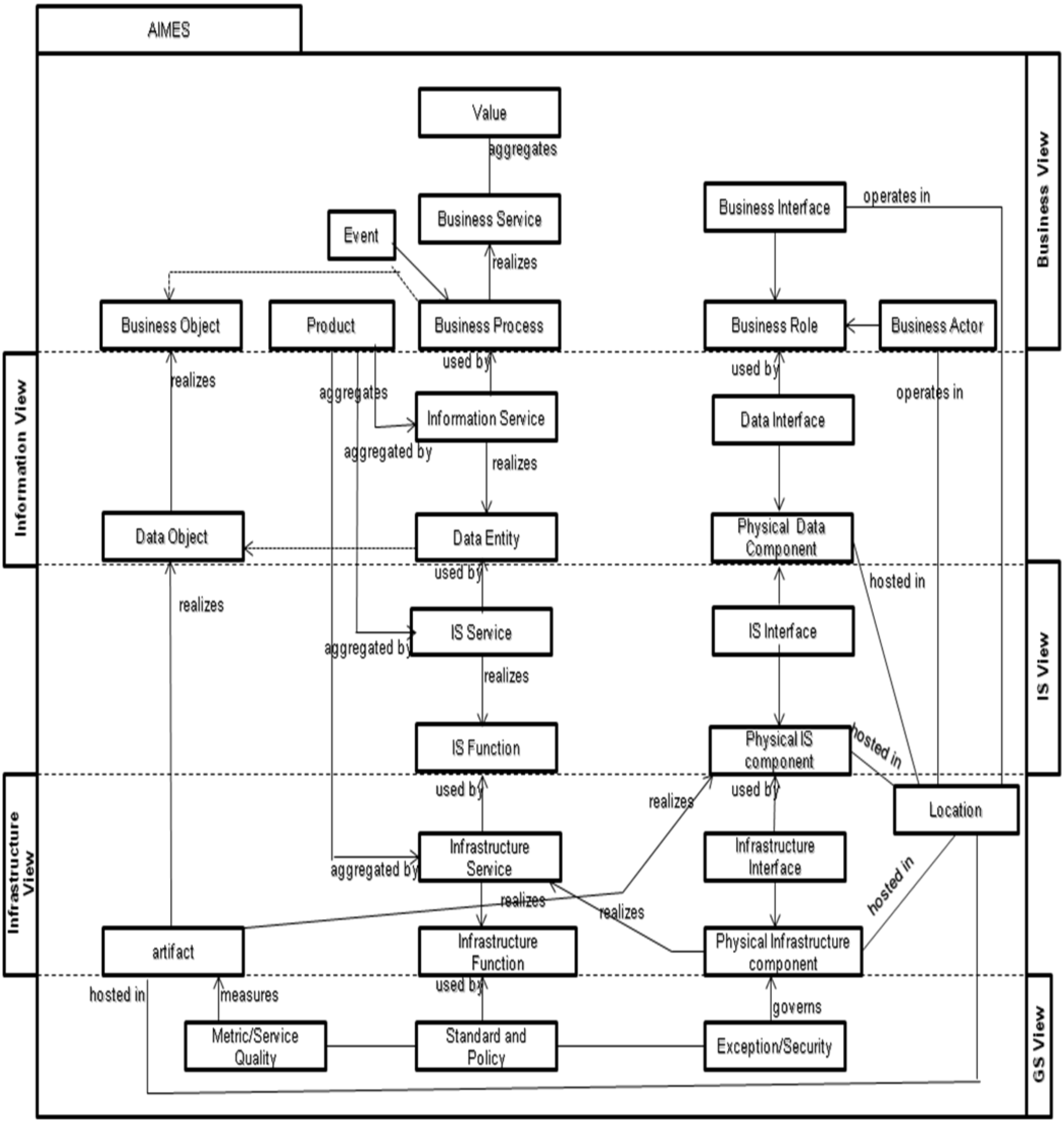


Figure 4.16 AIMES Integrated Metamodel for SME

The new metamodel introduced several new artefacts and clarify some ambiguous terms. It presents a more formal representation of the relations between fundamental components of enterprise architecture. It changed meaning for some artefacts in order to achieve a better stakeholder understanding and terminological harmony with other relevant standard.

4.1.7.1 Application to the SME: Donavan Integrated (Meta) Model

Figure 4.17 gives an example of a layered enterprise architecture description using services to relate the infrastructure layer, the application layer, the business process layer, and the environment. The customer and sales officer roles represent the client and the SME (Donavan), respectively. Invocation of the order request service by the customer starts the ordering process. The sales officer is informed whether the order is accepted, and, if so, receives a payment. Interaction between business processes and organisational roles is through business services. Thus, services connect the process architecture and the organisation architecture. Likewise, application services relate the business process architecture to the application architecture. The automated part of each business process is provided by an external application service. These application services are realised by application components. Finally, the technology layer consists of a number of infrastructure elements such as a HP Server and an application server, which execute application components and provide services to the application layer.

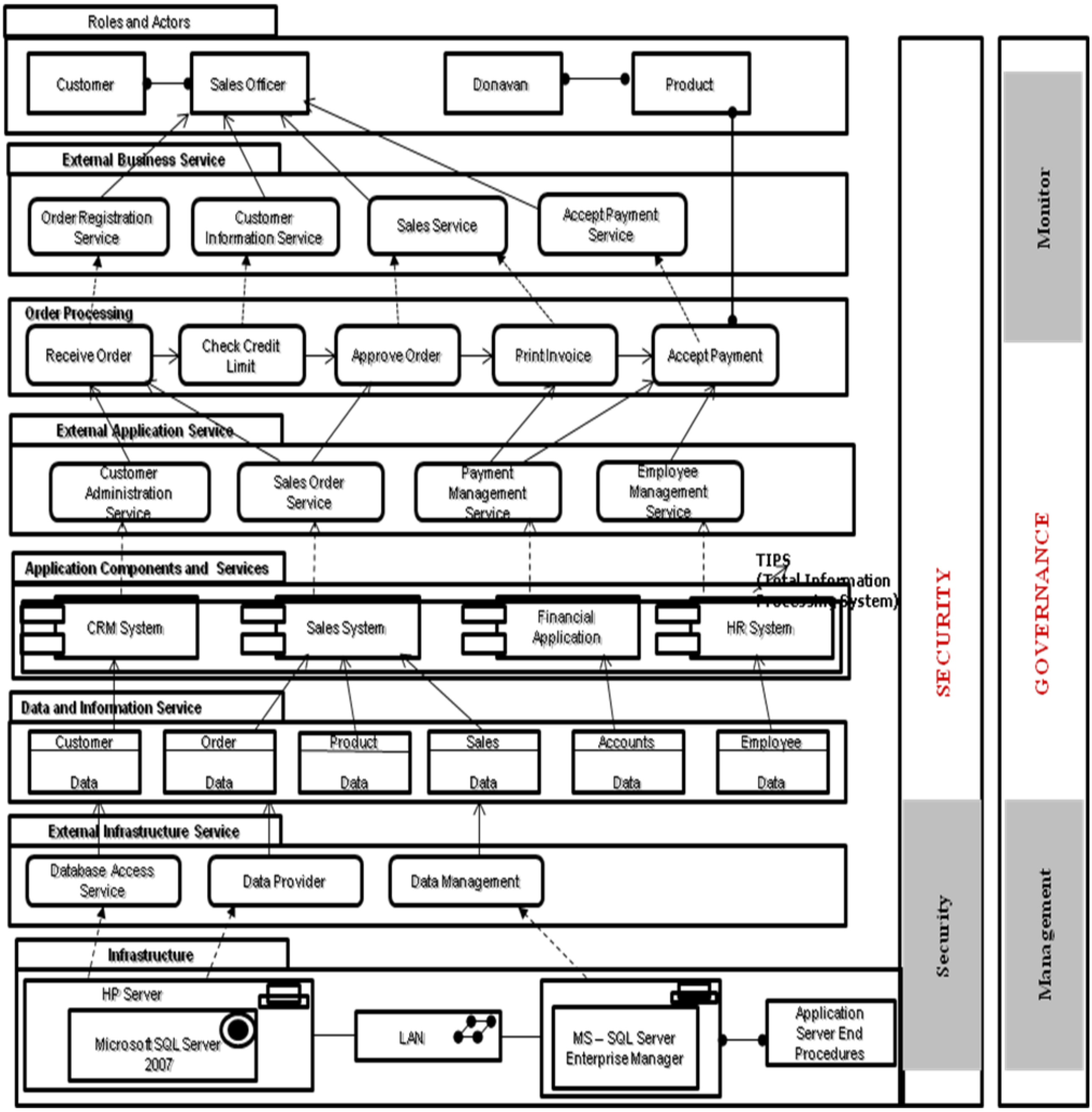


Figure 4.17 Integrated Metamodel of AIMES views in Donavan (by Author)

4.1.7.2 Analysis of the Integrated Model for Small and Medium-Scale Enterprise Systems

The Integrated Model for the SME (Donavan) shows the actors and their roles in the SME. It also shows the core business processes, the information that is needed by the business, the information system that supports the processes and the technology that supports information system and information. It provides traceability from business function, to a business process owned by that function, to a location of the business process, to a supporting application the process needs, and finally to the technologies that support that application.

4.1.7.2.1 Realized Values

Analysis of the architecture shows that some values were realized from the new architecture. These include;

i. Technologies were retired
   - Database server was consolidated in SQL Server thereby reducing data redundancy and technology cost by reducing the number of information systems and infrastructure that supports them.
   - An integrated System was implemented to be used by the enterprise. This reduced overhead cost of supporting and maintaining disparate systems.

ii. Reduced IT Skill Set
   - Dbase developer and MS-Access expert – due to limited resource in the SME, they migrated off Dbase V and MS-Access into the SQL Server reducing the number of IT personnel needed. This in turn reduced overhead cost for the CEO.

iii. Reduced Development Cost
   - Provided common infrastructure for development (IDE and .Net framework)
   - Object oriented focused enabling code reuse and implemented in windows platform.

iv. The Order processing time was reduced from 10 minutes to 5 minutes, saving processing time.

v. Standardized business processes and streamlined operations in the SME thereby reducing the number of business processes and also reduced overhead cost.

4.1.8 Example of Derived Relation – Based on Customer

An actor plays a role; a system fulfills an application component, an application components accesses data.

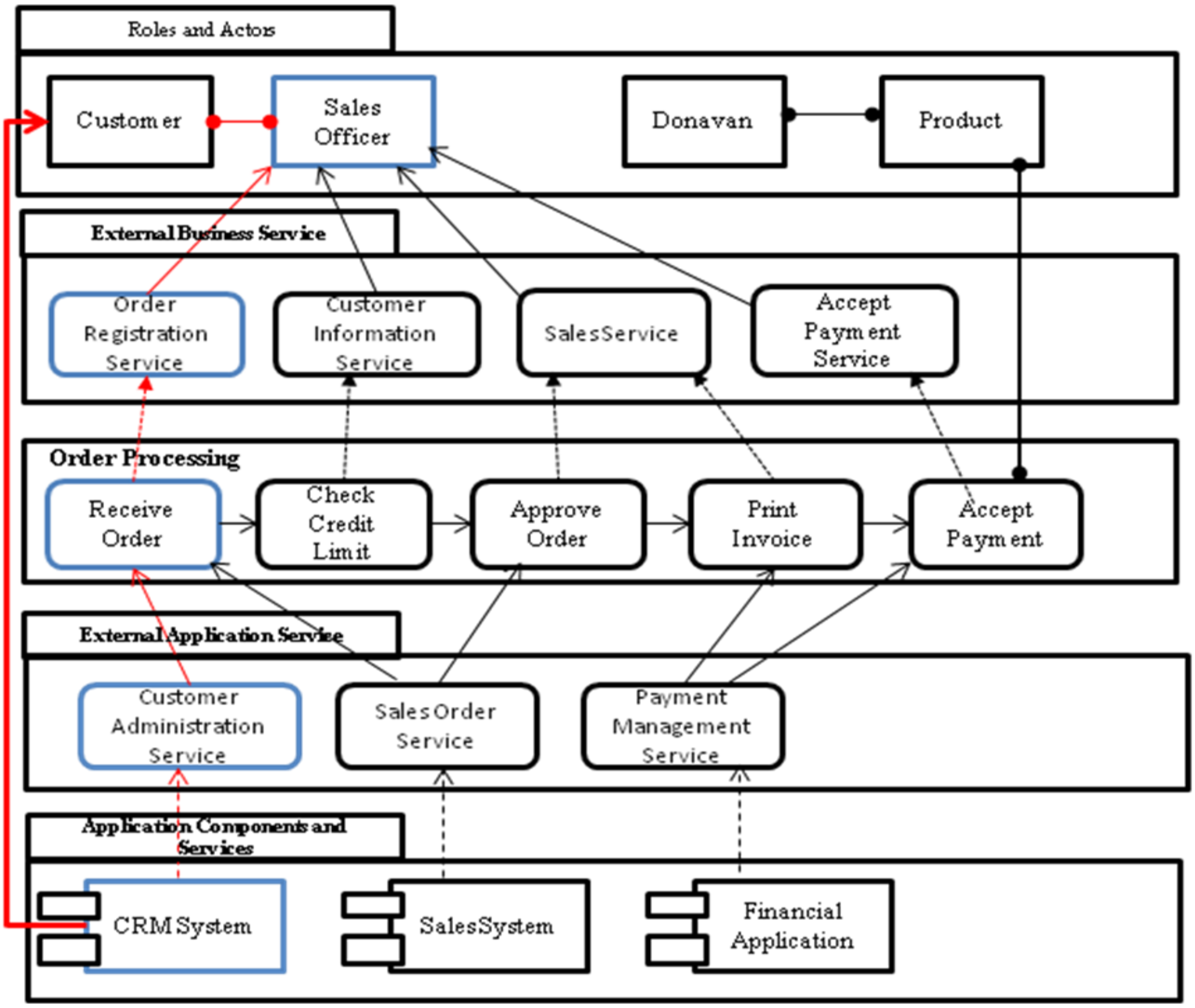


Figure 4.18 Example of a Derived Relation

Example: Donavan Layer Relationship

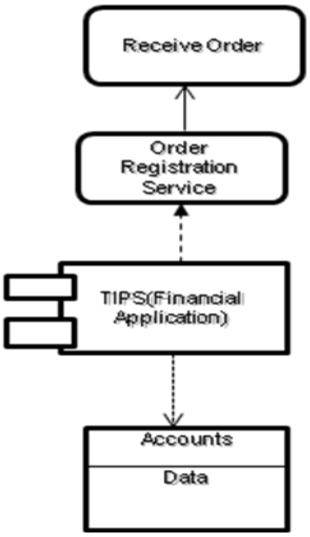


Figure 4.19 Example of Describing Relationships

4.2 AIMES Business Drivers and Principles for the SME

4.2.1 Business Drivers

Through understanding the business drivers, we can put in place architectural guidelines which will shape decision making in the SME**.**

i) Business Influences Architecture

Architectures must respect business goals and values. These goals will shape the way in which we will make decisions when selecting architecture options and components. Our choices may be very different depending upon whether we want to optimize cost or time to market. The business drivers of AIMES are reducing cost and complexity of information system, efficient information system, IT/business alignment.

ii) Technology Capabilities Influence Business

The reverse side of the coin is that emerging technologies can create business opportunities. By considering innovations relevant to our industry, we can identify opportunities for business innovation that can be supported by the technology. Sometimes the new approaches can offer whole new business models. Technology capabilities include new platforms (software, hardware, networks, and internet).

4.2.2 AIMES Principles

i) **Business Principle:** This principle states that "All employees have to access data, without special software, all over the world".

ii) **Information Principle:** This principle states that "The owner of data is responsible for the actuality of information".

iii) **Information System Principle:** This Principle states that "Components must be reused".

iv) **Technology Infrastructure Principle:** This principle states that "Security profiles and business rule will be implemented in the network layer".

v) **Governance and Security Principle:** This principle states that "The infrastructure will be centralized managed, application support will be decentralized organized and only owners of the data can change or delete information".

4.3 AIMES Enhanced Methodology for Enterprise Architecture Development (Method) in SMEs

The methodology consists of discrete stages that define how to create and use enterprise architecture in SMEs. It provides the principles and practice for creating and using the architecture description of a system. This method provides a step-by-step 'Enterprise Architecture Design methodology', to guide the construction of an integrated enterprise-system using the business process vision and capable of dynamic evolution depending on necessities and objectives. The enterprise integration approach that is proposed following the AIMES architecture, from a practical application point of view can be summarized as follows and depicted Figure 4.20.

Step1: Establish architecture requirements: Define the purpose and requirement for the EA. This can be a set of specific questions on critical areas. This includes Mission, Vision, objectives, values, organisation, Function and stakeholders. Eg. What technology should we use today for Web application and databases?

Step2: Design the Architecture Views: This includes designing business view: Define business capability, business processes, and business services. Information View: Define data capabilities; data object models, Data flow diagram, ERD Model, physical model, database design.

Step3: Design Services and System: Define Application capabilities, Application diagrams, Object interaction, Classes, Interfaces.

Step4: Design the metamodels for defined architecture view based on known concepts.

Step5: Design Infrastructure and Resource View: Define the technical reference architecture. Infrastructure and resources capabilities, technology support models.

Step6: Integrate the Architecture on key artifacts and services given and received.

Step7: Enforce Architectural Control: Choose key technologies and standardize.

Step8: Validate and Execute Architecture: Testing and Evolution of the Enterprise architecture with users and stakeholders.

Step 9: Evolution and Continuous Improvement: Identify the desired enhancement to business and then final dissolution where necessary.

The methodology is as represented in Figure 4.20.

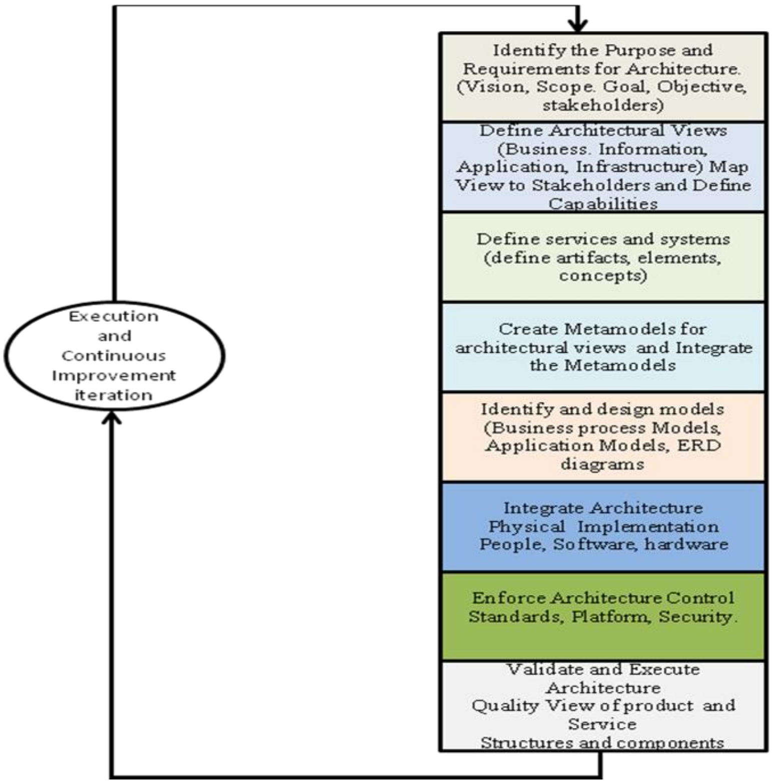


Figure 4.20 The AIMES Architecture Design Methodology (by Author)

The architecture design methodology (Step by-step approach) identifies the process that must be followed in order to build enterprise architecture model in small and medium scale enterprise (SMEs). The methodology structures architect's thinking by dividing the architecture description into domain, layers or view and offers models – typically matrix and diagrams for documenting each view.

4.3.1 AIMES Abstraction Levels – Essential Dimensions of AIMES in the SME

Abstraction allows a consistent level of definition and understanding to be achieved in each area of the architecture. It is especially useful when dealing with large and complex architectures, as it allows relevant issues to be identified before further detailing is attempted. This approach is found in most architectural approaches including The Open Group Architecture Framework (TOGAF) and the Zachman Framework for Enterprise Architecture. The Integrated Model for Small and Medium Scale Enterprise Systems (SMEs) defines fives Levels of abstractions.

i) The **contextual phase** answers the question: What ICT enabled enterprise (i.e. its business organisation, information/knowledge organisation, information systems and technology infrastructure) must be realised? The deliverables include:

· The vision, strategy and policies concerning the ICT enabled enterprise.

· The scope of the enterprise that must be (re)designed.

· The stakeholders of the design and the viewpoints they must assess.

· Requirements and constraints concerning the structures and qualities of the ICT enabled enterprise.

ii) The **Conceptual phase** answers the question: What capabilities are needed to support the business? This is where requirements and objectives are analyzed and elaborated. It ensures that all aspects of the scope are explored, relevant issues identified and resolved, without concern over how the architecture will be realized.

iii) The **logical phase** answers the question: How is the ICT enabled enterprise realised? What applications and tools are needed to support the business? The deliverables include:

· A design of the behaviour and appearance of the business organisation, information/knowledge organisation, information systems and technology infrastructure of the enterprise and the products and/or services these organisations and systems delivers.

· A design of the co-operation and construction of components of these organisations and systems within the enterprise. In fact the components are roles performed by people in the business and information/knowledge organisation, by software components in the information systems and by software and hardware components in the technology infrastructure.

· The transformation stages of the IT enabled enterprise as base for a migration or transformation plan.

iv) The **physical phase** answers the question: With what is the IT enabled enterprise realised? What technology and people will we use to enable the applications and tools? The deliverables include prescriptions for:

· The capabilities of human resources in the organisation and the software and hardware components in the information systems and technology infrastructure.

· The development and realization methods.

v) The **Execution and Continuous improvement** answers the question: How good is the IT enabled enterprise that will be realised? What will the standard and procedures to guide the implementation? The deliverables include:

· Views that represent the structures and qualities of the organisation and systems for assessment by the different stakeholders.

· Reports about the assessment.

4.4 Extracted Classification Model for Business Applications in the SME

Classifying applications helps to understand the contribution expected from current and future information systems in the enterprise. Application classification models provide instruments for balancing the application portfolio and the life cycle of applications.

Table 4.12 AIMES Application Classification Criteria

Model	**Classification criteria**	**Categories in the model**
Proposed model (AIMES Application Model)	- Purpose of the use of the application - Business processes	- Back end applications - Front end applications - Engineering applications - Management tools - Communication applications

Based on the analysis of existing application classifications, we enhanced the models, integrating categories to design our own business application model. The model will be used when analyzing the development of information system view from the application driven point of view for enterprises.

4.4.1. Classification Principles in Proposed Classification model

In the proposed model, classification is based on the purpose of the application itself, the business processes of the enterprise and it is independent of organizational levels. Applications are classified as Back end applications, Front end applications, Management tools, Engineering applications and Communication applications. This classification is illustrated in Figure 4.21a.

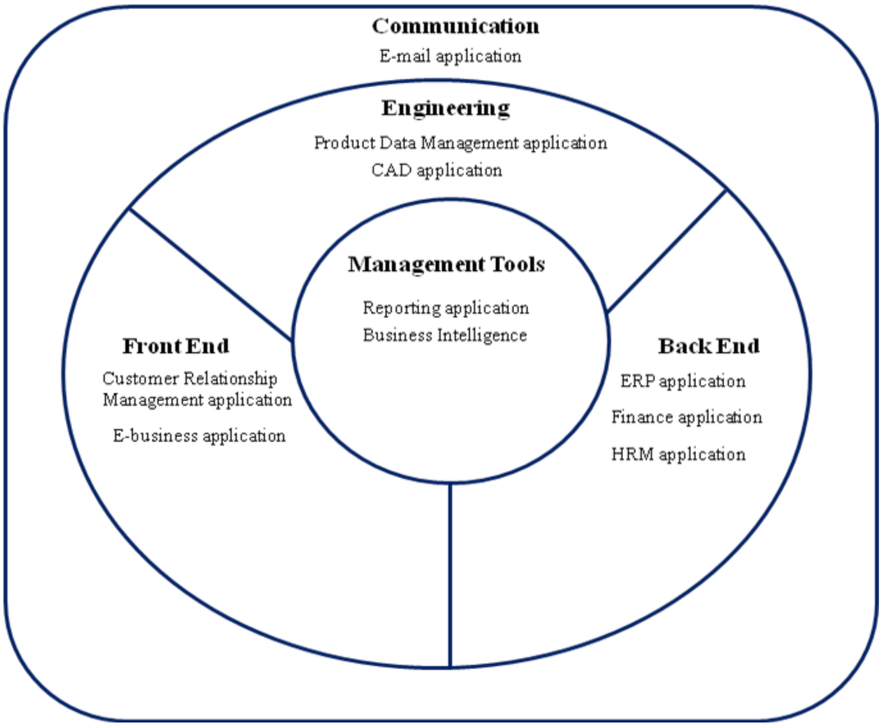


Figure 4.21a Proposed AIMES Application Classification Model

Brief Definition of Model Elements

a. Back end applications

Back end applications are transactional applications that perform transactions and operations, which are not directly in the customer interface. Such transactions are for instance order processing, inventory management and accounting. Back end applications for instance form the backbone of an organization's business applications. Back end applications include for instance ERP, finance, human resources management and payroll. If the enterprise does not have a comprehensive ERP system, these functions can be covered by several independent applications. Although ERP systems have clear advantages and an ERP package is the dominant strategy in enterprise to gain business process integration, it is not always obvious that a company should implement a comprehensive ERP package. An alternative strategy would be to use best of breed packages for specific purposes (Light et al., 2011). The main reason for this alternative is to achieve flexibility in the information system view. In several cases, this may be a better option, because case studies in literature highlight the unresolved inflexibility of ERP applications (Davenport, 1998; Kumar & Hillegersberg, 2000; Hagel & Brown, 2001).

b. Front End applications

Front end applications are used in customer interface. Front end applications are classified applications, which are used in customer interface or are utilized to create or distribute value-added services for customers. Front end applications include for example Customer Relationship Management (CRM), e-business and extranet applications. Value-added services created and distributed with front end applications are for instance electronic user manuals, customer drawings and product service history information databases. In companies dealing directly with consumers or end-users frond end applications may include Point-of-Sale or shop management applications. As ERP applications get more and more comprehensive in some case they have features of front end applications, although ERP applications are classified as Back End applications in this classification model.

c. Engineering applications

Engineering applications are used for engineering and R & D. These include for example Product Data Management (PDM) and CAD. Traditionally PDM and CAD applications are linked tightly together and currently also PDM and ERP are often linked (Soliman et al., 2001), but this creates problems relating to data overlap (Miller, 1999). Engineering applications are characteristics for manufacturing companies. Especially these are used in manufacturing companies with engineering function, although engineering applications can also be used for design. Engineering applications are often used in companies that deal with complex product data. This does not mean that the product itself would need to be complex, but it may include a lot of product related information. Another type of use is companies producing drawings. These drawings may concern for example products, product assemblies, product design, lay-outs, system planning etc.

d. Management Tools

Management tools are applications used to support management activities in all levels of the organisation. Management tools are typically applications that do not make transactions, but collect, refine and distribute information to support management activities. Management tools include applications such as supply chain management, data warehouse, business intelligence applications, budgeting and reporting. These applications are often integrated with backend, front end and engineering applications and they collect data from these applications and refine it as useful information. A typical management tool is a data warehouse application which collects data from various ERP applications in the enterprise, refines it and then distributes information throughout the enterprise. This information can be for example financial figures from various business units. Another useful way to exploit data warehouse is to use it to distribute information in the supply chain. This means quotations, orders, stock count and work in progress. With data warehouse it is possible to achieve visibility through the supply chain without heavy ERP implementation. However, the difference is that data warehouse is not a transaction system, so it only distributes information but does not automate transactions in order fulfillment process as an ERP system would do. This is why data warehouse is classified as a management tool, not as a Back end application. As Inmon (1999) stated "Data warehouse solves the problem of integration, except it solves the problem only for informational processing. As appealing and as powerful as data warehousing technology is, it is not a solution for operational transaction integration.

e. Communication application

Communication applications are used for collaboration and communication within the company and with external parties. Communication applications include applications such as e-mail and groupware.

4.4.2 Mapping TIPS Module to Application Classification Model

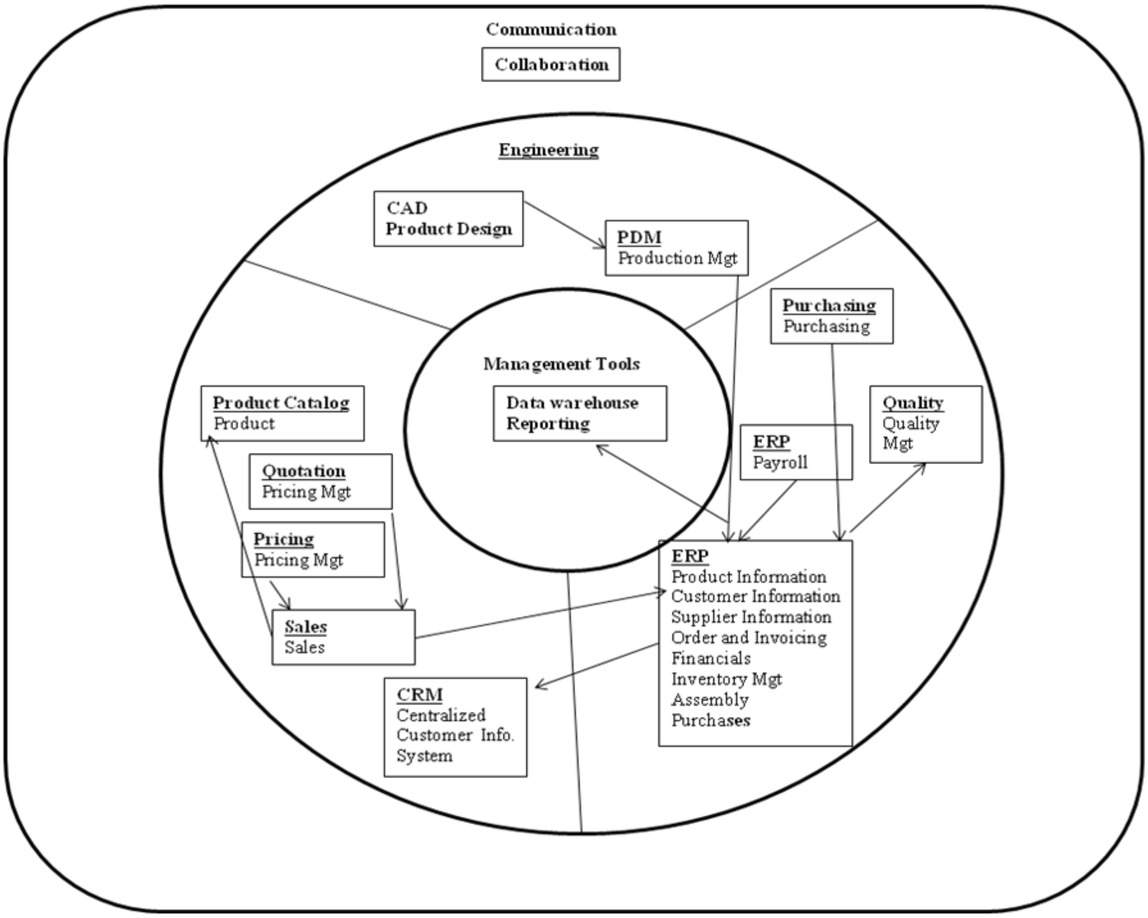


Figure 4.21b Mapping Extracted Modules to AIMES Application Model

4.5. Aligning the SME Using AIMES: Extracting the IT-Business Alignment Model

One test of the usefulness of enterprise architecture is its ability to assist with alignment, or to spotlight misalignment. In essence, if goals and measures are organized, managers should be able to see exactly how high level goals are implemented at successive levels of detail. In some cases, everyone will be happy with the iterative subdivision of goals, but in other cases, managers will probably find that goals become subtly distorted as they are implemented in lower level systems. While the enterprise architecture provides a tool that managers can use to see how goals and processes are implemented, throughout the organization, the managers themselves must decide if the various implementations actually accomplish what is intended by senior management.

Organizations are best aligned with goals and measures that determine if those goals are being met. Thus, creating a diagram that shows how a given strategy is to be supported by a specific business process, which in turn is implemented, in part, by software applications and databases, doesn't guarantee alignment. It merely guarantees that the person looking at the diagram knows all of the elements that are supposed to be coordinated to achieve the strategic goal. Even that, however, is a major improvement over situations in which managers are unsure what specific processes a given application is designed to support. Once one knows how the elements are intended to be aligned, one is positioned to use specific measures to determine if the processes, sub-processes, applications and databases are actually achieving their purpose. This is done by working top-downward and refining the goals and measures that will determine success. It begins with statements of what will constitute a success for the strategic initiative. That in turn, is used to determine what outcomes the process or processes assigned to that strategic goal must achieve. Sub-processes within processes are assigned their own specific goals. Applications designed to support sub-processes inherit their goals and measures from their processes, and so forth.

4.5.1. Aligning Business Processes with Information Systems

One of the objectives of this dissertation is to achieve better alignment of information technology with business objectives. The AIMES architecture will be used to align business processes fully with information technology. Figure 4.22 depicts the process of business process alignment with information technology. Information technology comprised of Information system view and Infrastructure.

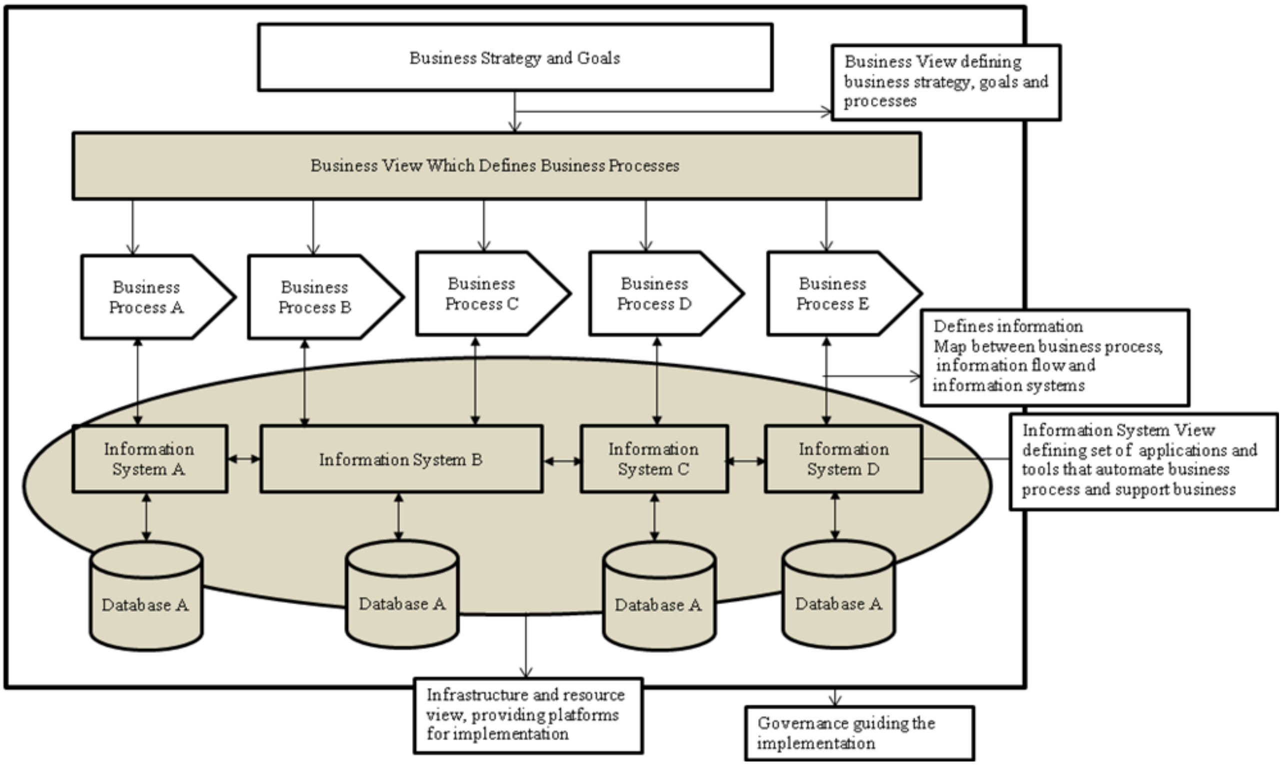


Figure 4.22 Aligning Business Processes with Information Technology

The business strategy and goals defines the vision and objectives of enterprise. This is transformed by the AIMES business view into business processes which defines the domain, relations and attributes as shown in Figure 4.22. The business process is now designed and map to the information system view which automates the business process and supports the business strategy and goals. The infrastructure and resource view supports the information system view by providing the people, platform, application program, data storage, network connectivity that supports the information system development and implementation. This supports the business process and achieves a total integration – alignment of business strategies transformed to business processes with information system supported by infrastructure. This is one area that most existing enterprise architecture has failed to give a clear definition.

Figure 4.23 provides a more detailed and concrete example of how processes are aligned with information systems. In this case, we took a slice across the organization and identified four specific business processes: an Order Process, an Assembly Process, a Delivery Process and a Billing Process. Each of these processes is supported, in Figure 4.23, by a software application, and these, in turn, are supported by databases. This view of the relationships among processes, applications and databases is a natural outcome of

AIMES enterprise architecture. Too many organizations, without a comprehensive architecture, associate applications with departments or specific functions. The keys to a well-organized architecture are (1) goals that are tracked through the layers of detail, and, (2) a set of business processes that provide an overview of the work and the outputs of the organization. Together, this information clarifies how processes link, directly or indirectly to customers, and how they achieve specific organizational goals.

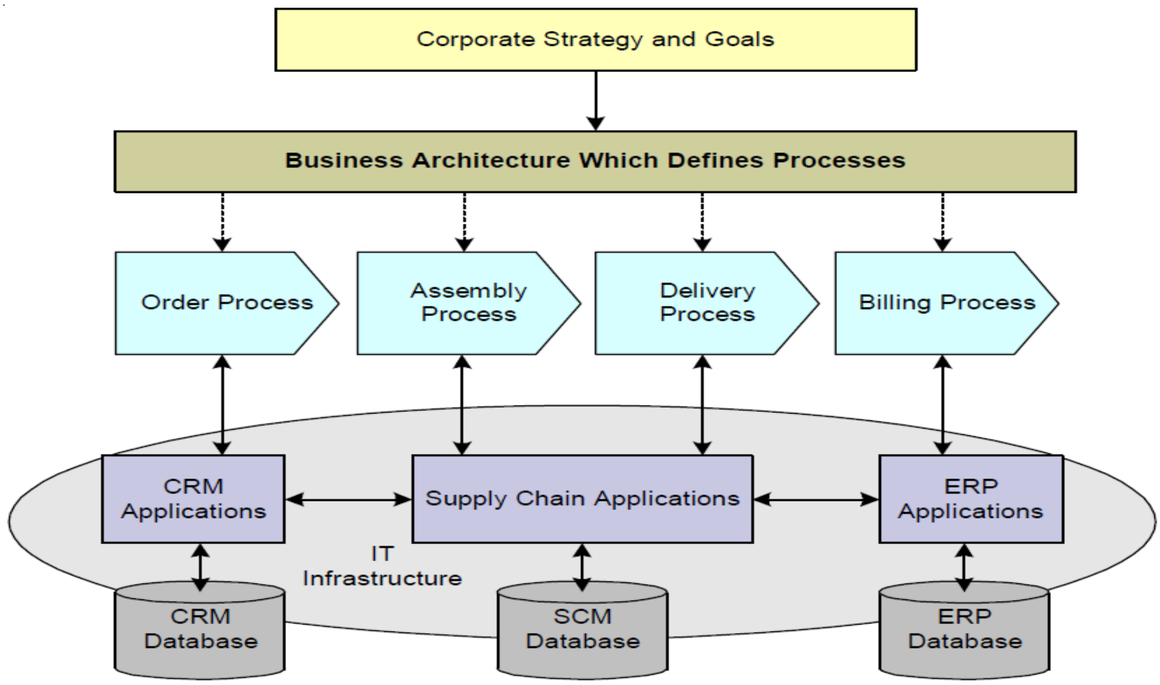


Figure 4.23 Aligning processes, applications and databases

4.6. Practical Application of AIMES in an SME: The Case Study

The **AIMES** architecture was applied to an SME in order to evaluate its applicability as an enterprise-level architectural design tool. In our case, the enterprise is an SME (Small and Medium Enterprise) – A Tile Manufacturing.

The Donavan Tiles Ltd (DT) is a subsidiary of a multinational. It provides internal services like procurement logistic, production service, financial administration and human resource management, as well as external services for customers and organisations such as sales, supply and distribution. Donavan tiles may be privatized in the near future, which has several consequences. The IT provisions of Donavan tiles must be prepared for the future. Even stronger, the management of DT regards IT as an enabler for offering new products and services to the customers in a timely way.

This case study shows an enterprise-level architectural design using AIMES that relates the organisation of a business with information technology (IT) in order to assess the feasibility of the strategy of Donavan Tiles. In particular, AIMES has been used to assess whether Donavan has the potential to adapt to new challenges in a cost-effective manner.

4.6.1 Present Procedure

Presently, the company has information systems that handle the different core processes. As the systems have been developed by or for specific departments, the systems would often do exactly the same as the manual steps used to giving a very narrow scope. The resultant systems are generally independent of others as shown in figure. Each system is on its own solving the need of a particular process for which it was developed and shares no data or resource with any other system thereby making data and process exchange difficult because this can only be done manually. Figure 4.20 shows the current business process.

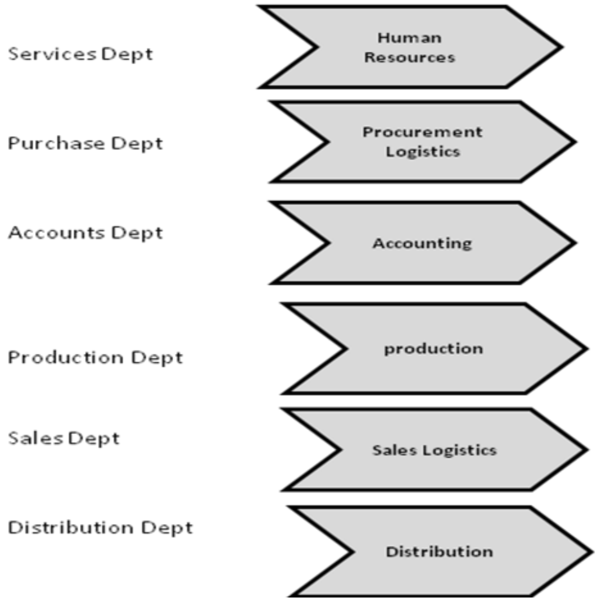


Figure 4.24 Current Business Processes and ES Business Cycle

Table 4.13 Current Business Processes

Procurement Logistics: is the process of buying and obtaining the products worldwide. The process involves several activities such as determining the markets needs, locating supply sources, evaluating and selecting one or more suppliers, choosing a buying method, monitoring the purchase's status, receipting, evaluating and storing products.
External Accounting: involves key areas such as accounts payable and accounts receivable, which are closely linked to cash management and forecasting. As in any other company, this is the vital business process in Donovan
Production: the use of the raw materials to produce finished products
Distribution: refers to replenish products (only tiles in containers) into regional distribution warehouses from the central warehouse.
Sales Logistics: involves three key sub-processes: customer request for quotation (RFQ) processing, customer sales orders (CSO) processing and delivery processing.
Human Resources: manages people and resources of the enterprise

The process was characterized by the major systems (Production system, purchase system, sales order system, inventory system, finance system) with manual or inexistence interfaces. The systems developed "in-house" by the company did not provide support to actual business needs. The company's business process is not integrated and as such does not have any Integrated Enterprise system.

4.6.2 The Baseline Architecture – Island of Automation

The Baseline architecture – Islands of Automation, presents a conceptual view of the present environment. The current infrastructure, databases and application systems have not been driven by an enterprise-wide architecture and exhibit lack of commonality, interoperability or portability, as one would expect. Such systems are described as "Islands of Automation" and "Stovepiped" – two metaphor that refer to their fragmentation and independence both in lack of commonality and in interoperability. The structure of the software that runs on the hardware is not guided by any perceived enterprise-wide guidance as shown is figure 4.25.

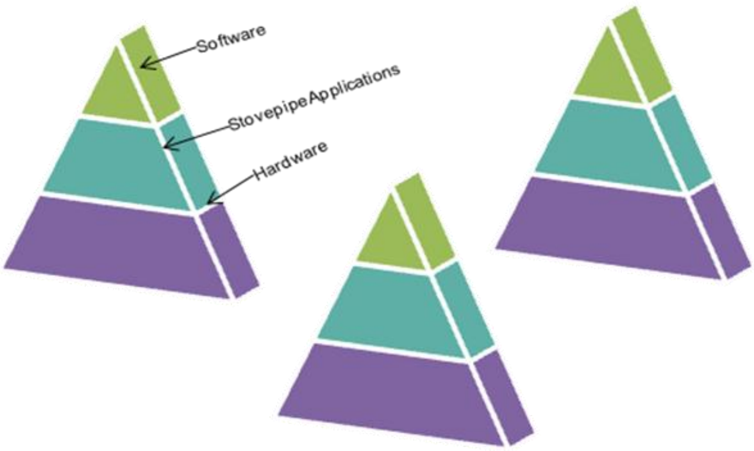


Figure 4.25 The baseline architecture – Islands of Automation

Although the company has made significant effort through some modernization effort, the environment remains plagued by components that are non-standard and not interoperable. As a result, end-users find it difficult to identify and locate information of interest, and collaborative processing is severely limited. Transaction processing systems use non-standard approaches and user interfaces, increasing the training and maintenance burden for the company.

4.6.3 Flow Diagram of the Present Model

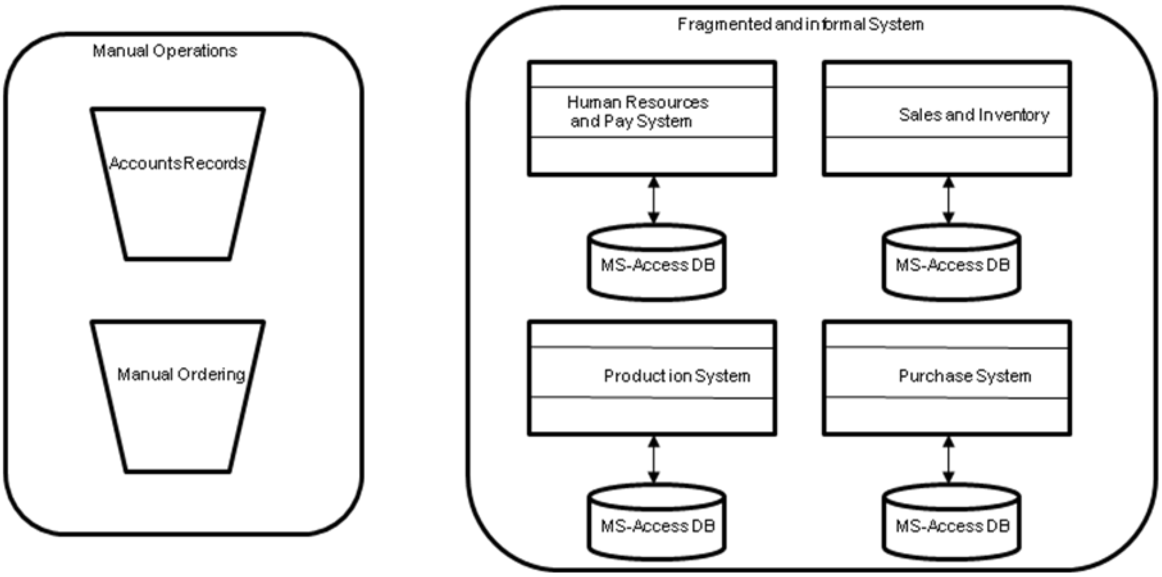


Figure 4.26 Information Flow Diagram of the Present Procedure

4.6.4. Business Requirements and Future Directions

The current IT policy is based on supporting Donavan Tiles processes with dedicated IT solutions. The pros and cons of a “make-or-buy” solution are assessed carefully. However, less attention has been paid to the integration of these processes, resulting in IT islands supporting very specific services. IT is not explicitly used to enable new forms of organisation or processes.

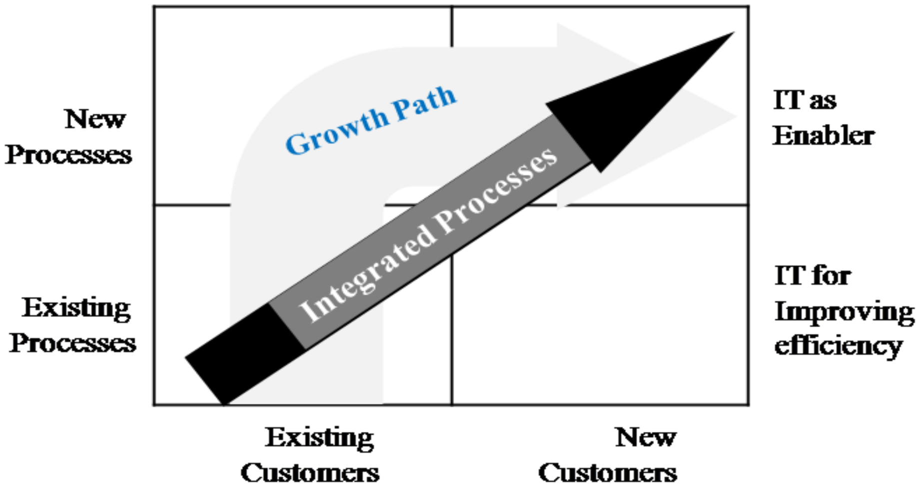


Figure 4.27 The growth path of Donavan Tiles

We defined a growth path comprising of two stages (Figure 4.27):

i. Offer new services to the existing customers;

ii. Offer new, integrated services to existing and new customers.

4.6.5 The Solution: Our Approach

Since the organisation of the business and the IT goes hand in hand, the consequences for acquiring future directions cannot be analyzed in isolation. For this reason, the feasibility study has been conducted that uses AIMES to relate business issues with IT issues.

The desired organisation, which is partly reflected in the current IT solution, is used as a starting point in the assessment of the requirements. The following phases have been followed:

1. Envisioning the desired situation;
2. Devising alternative solutions;
3. Evaluating the alternatives from a flexibility and cost-effectiveness point of view;
4. Elaborating one alternative, provided that future directions can be reached.

4.6.5.1 Envisioning the Desired Situation: Complete Enterprise Integration

In order to design an integrated system for the implementation of AIMES, the enterprise must be integrated first. This was achieved using Business Process Modeling. The basic concept is that any organization can be thought of as operating or behaving as a number of interrelated processes. A process is a set of inter-related work activities. These activities requires specific inputs from one or several “originators,” perform some specific value-adding tasks on those in-puts (some kind of transformation, integration, customization, etc), and deliver specific outputs to one or several receivers.

To design an integrated enterprise is to show how the elements of the enterprise are working and interrelating in order to achieve the Business Objective of the Enterprise. This is modeled using the AIMES architecture design methodology.

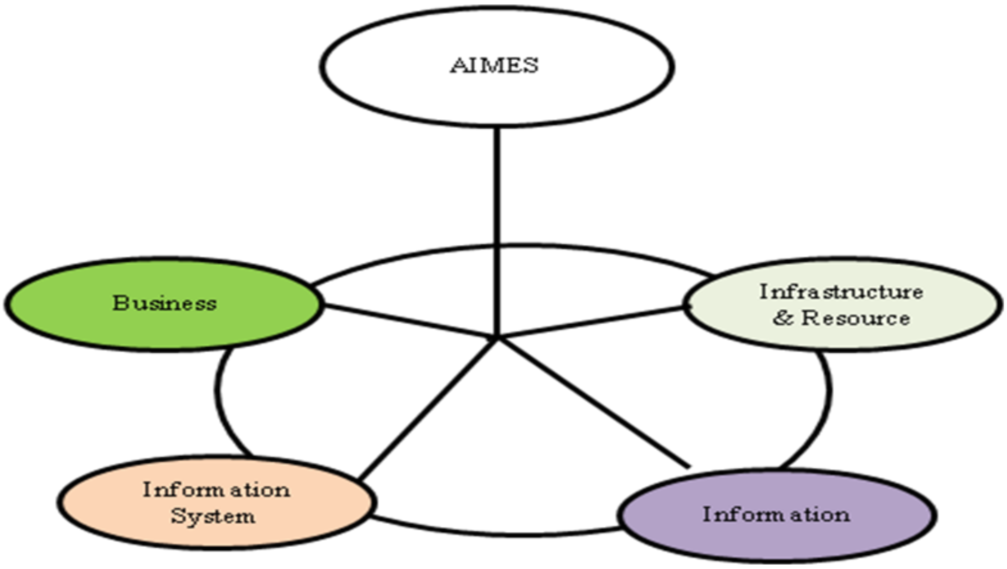


Figure 4.28 The Integrated Enterprise

4.6.5.2 Alternative Information Systems Solution

The next stage is the enterprise-level design of the information systems that support the designed business roles in their business processes and information exchange and storage. Two alternatives were conceived that are primarily evaluated in relation with the Business and Environment (BE) and the Information and Knowledge (IK) architecture areas with respect to enabling the flexibility of the organisation and the total cost-effectiveness.

The two alternatives can be summarized as:

1. ERP (Enterprise Resource Planning)

An ERP package is comprised of several modules that handle specific services. The emphasis lies on production system support rather than customer support.

2. Custom Integrated Application

The custom integrated application uses a single system for management, sales, stock, Human resources and customer services.

4.6.5.3 Evaluating the Alternatives from a flexibility and Cost Effectiveness Point of View

The pros and cons of these alternatives are shown in Table 4.14

Table 4.14 Evaluating Alternatives

SNo	Alternatives	Flexibility	Cost-Effectiveness
1	ERP	Integrate is easy (+) Fixes business organisation (-) Unsuited for IT enabling (-) No complete coverage (-)	Expensive implementation (-) Must be interfaced with other IT applications for Complete coverage (-)
2	Custom Integrated Application	Tailored to Donavan Tiles needs (+) Integration is relatively easy (+) Required Functionality not fully guaranteed (-)	Well suited for enabling IT (+) Inexpensive integration operationally (+) Expensive Integration Initially (-)

4.6.5.4 Elaborating Alternative 2: Custom Integrated Application

By taking the pros and cons into account, it was decided to elaborate the second alternative. The consequences for purchasing this alternative were evaluated mainly at the Information System (IS) and Technical Infrastructure (TI) architecture areas. However, the impacts of technology-driven solutions were assessed again at the business level to assure that the business requirements set out by the management are satisfied. One consequence was clear from the outset, namely that the Donavan tile has to adapt to custom integrated application. This is not necessarily a bad thing, the DT organisation must be prepared to adapt anyway, not only by customer forces, but also by new technologies.

The quality attributes that are of interest in the IS and TI architecture areas include flexibility, security and performance. Notice that cost-effectiveness is a less important quality attribute since the decision to use cost-effective custom made application was already made at the BE and IK architectural levels.

The Information System concept can be summarized as follows:

i. The custom designed application must support preferably a three-tier-architecture, i.e., separation of data, business logic, and user interface (flexibility);
ii. a middleware will be used to exchange information between the custom designed and to synchronize business processes (flexibility, security, and performance);
iii. Specific custom application requirements:
iv. Custom application (integrated enterprise information system): three tier architecture;
v. Donavan-Front-End (customer service point, help desk)
vi. Donavan-Customer Relationship Systems (very specific, single services): (de-facto) industrial standards;

By using custom designed applications based on three-tier-architectures and by using de-facto industrial standards and technologies, the requirements imposed from a business perspective (e.g., flexibility and cost effectiveness) can be satisfied.

4.6.6 Design of the Business View for the SME

The goal of this phase is to relate business processes with IT support to obtain an overall view of the desired situation. The following steps have been carried out:

i. Identifying organizational units and other actors (internal and external);
ii. Identifying their roles and interaction in the business process;
iii. Identifying necessary information for each role and the processes to provide this information.

This is a high-level logical design in the BE and IK-area of AIMES. The result of step 1 is reflected in the process map (Figure 4.29).

A. Modeling Enterprise Business Processes with AIMES Architecture Design Methodology

Based on the AIMES design methodology, we model the enterprise business processes.

(i) Objectives

Donavan's enterprise architecture objectives are;

i. Improve the relationship between customer and supplier
ii. Minimize operation failure
iii. Have better management control

iv. Improve planning process
v. Improve IT to Business Alignment
vi. Reduce Complexity by establishing a process that is focused on building, maintaining, acquiring and retiring technology

B. Building the Enterprise Business Processes

A new process map was built for the Tile industry showing improvements in the existing process with some new processes which will be introduced in the To-Be process. The tile industry is made up of core eight (8) business units that made up the macro processes in the enterprise as depicted by the large arrows in the process map and are listed below:

Table 4.14 Processes organized by Business Area of the Tile Industry

S/N	**Business Unit**	**Process1**	**Process2**	
1	Production Management	Program production cycles	Management of the manufacturing process	
2	Human Resources	Personnel Administration	Manages staff recruitment, manage staff.	manage staff education etc
3	Warehouse management and Control	Manage raw materials, moulds and plates, packaging and finished products		
4	Finance	Management of company finances	Manages suppliers, clients, banks, taxes, travel expenses etc	Checks clients risk (Credit limit) for purchase order, manage insurance, control and follow up the non-payments.
5	Sales and Marketing	Management of clients	Order Management	Production promotion, promotion product order etc.
6	Distribution	Management of the delivery of finished products		
7	Risk Management	Management of customer's risk	Order management	
8	IT	Provision of information		

		technology, support for company's business processes		

In the table 4.15, elements of the business view were presented organized by business area. The core processes were described all for achieving the objectives and goals of the enterprise.

The structure of the company has 4 levels, corresponding to the CEO and sales manager at the top position, Head of production, Head Buyer, Head finance and Human resources manager. The other levels has supervisor for machining and pailer areas. The strategy is to produce with high quality that market demand.

Figure 4.29 shows the Tile Industry Process Map showing the relationship between the different business processes.

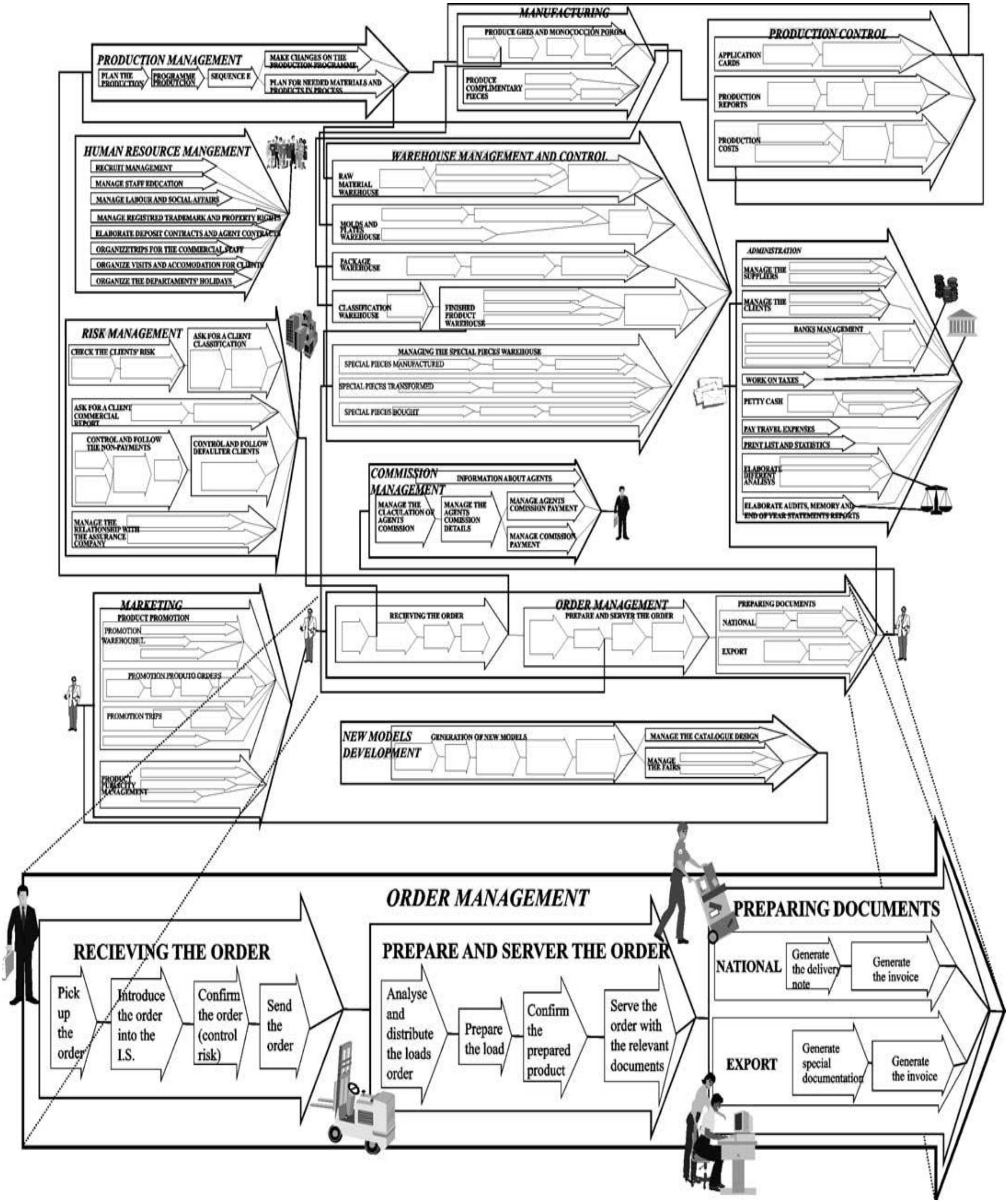


Figure 4.29 Donavan Tile Process Map

Figure 4.29 shows a process map diagram of the reference model of the tile industry developed in other to show the application of AIMES architecture. The Order Management of the Tile Industry was expanded to show how a process can be improved because it cuts across almost all the units in the business process. The big arrows show the tile industry macro-processes. Each macro process is divided into micro-processes, activities, tasks and so on until getting down to the lowest level, that is, an event produced by a human action or a machine. Every task of the model has inputs, controls and resources (Table 4.16.).

In addition to the operational part (activities realized), the model shows the information, the resources and the decisional alternatives for each activity within the process. Decisional alternatives represent the different ways of doing an activity and their election depends on the established goals for the process, according to the strategy and the politics of the company (see example in table 4.16).

Table 4.16 Example of a task

Introduce the order into the information system (IS)	
The aim of this activity is to introduce the order in the IS Clients that place orders have already been introduced by the commercial area, who in turn, have previously accorded payment conditions, sales conditions and prices with the client. Moreover, it has asked the risk area for client risk information.	
The order can arrive from two sources, by phone call or fax, whether coming from the client or from the commercial agent. Independently of the source from where the order arrives, their reception is centralized in the sales department.	
Inputs	Client order (telephone, fax, other media) Items information (code, size, quality etc) Client information (fiscal data, delivery address etc) Additional information of the client and/or the order provided by the commercial agent.
Outputs	Order
Controls	Catalogue Prices Enterprise politics IS capacity
Resources	Sales area staff IS, telephone, computer etc.

Someone has to take decisions. The process/function matrix should be completed in order to know who is the owner of every process (see Figure 4.30).

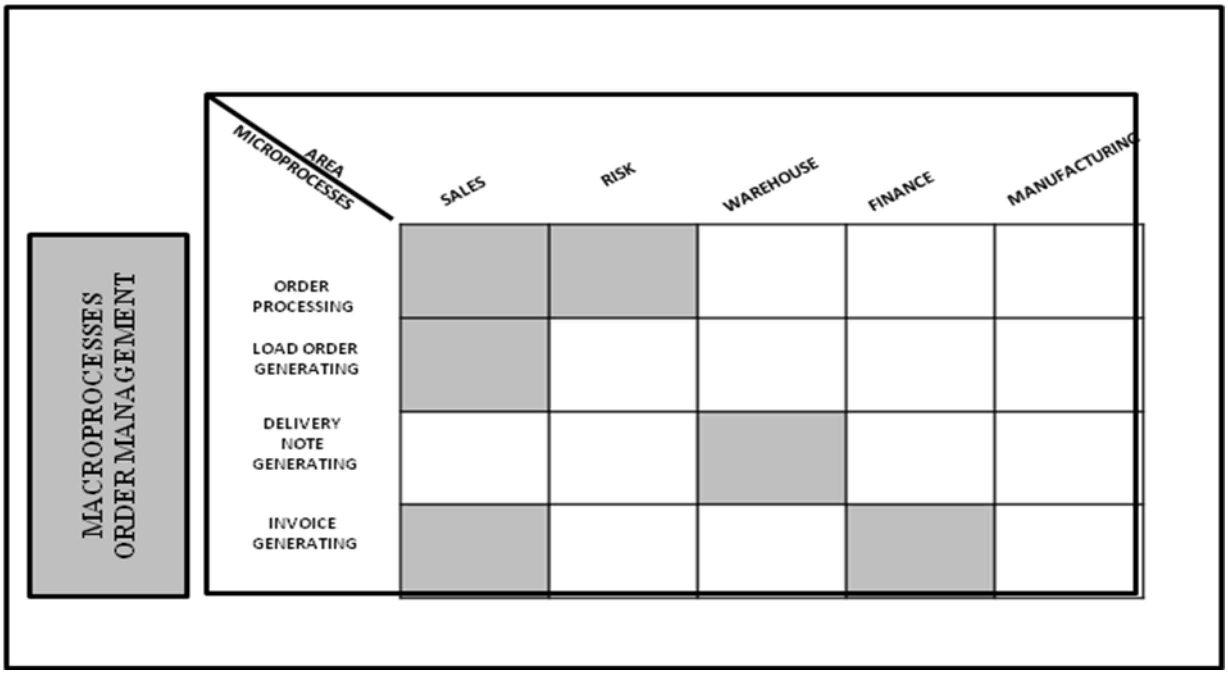


Figure 4.30 Process/function Matrix

In addition, the Industry process Map can be improved by coloring the activities carried out by personnel in the same department with the same color. In this way, the figure will represent the horizontal view of the enterprise by means of the departments.

Table 4.17 Example of decision alternatives

Decision analysis			
Situation 1	The client exceed the risk		
Decision 1	Decision 2	Advantages	Disadvantages
Introduce the order into the IS and wait until the commercial for director's authorization	Reserve demanded items	It will not load orders to high risk clients without a previous control by the commercial director A preservation is made and can be confirmed after the authorize	When a reservation is made, the available products will be kept for no risk client only
	Do not reserve demanded items. They are kept only as	Product that may not be served will not be reserved.	When the commercial director authorizes the order,

	information	No risk client orders can be places. It will not load orders to high risk clients without a previous control by the commercial director	it is possible that the available product item is not available
Do not allow the introduction of order into the IS			The commercial agent must find an alternative way to register the order which will generate duplicity.

This reference model will facilitate the business process design project in a tile industry. It will reduce from four months to one the time required to carry out the whole study: the analysis of the AS-IS and the improvement proposals (TO-BE).

C. Implementation

As a result of applying the business process design in the SME, a new process map was designed. The next step is to establish and to control the migration of the old process (the AS-IS) in the new processes (the TO-BE). Table 4.18 shows a template to begin the AS-IS/TO-BE analysis. This template is also appropriate for showing small improvements in the process since the AS-IS and the TO-BE activities will coincide, although they will be carried out differently. A significant re-design effort may result in a completely different set of activities, in which case another template model would be needed to show the process result regarding AS-IS.

The gap + restrictions column shows the differences between current work performance, desired work performance and the restrictions that exist to achieve the latter. In this way the TO-BE enterprise implementation may be broken up into series of co-ordinated projects', that will be set in priority and if the enterprise already exits, these projects will form the transition path from AS-IS to the TO-BE enterprise. Traditional cost/benefit analysis is a very useful tool to define the priority of the projects within the integration program. However, organizational, technical and operational aspects should be considered. Therefore, AIMES can be used as the basis for feasible enterprise project identification, execution and management, all of which are within the financial, physical and economic capabilities of the enterprise and the requirements of the conceptual phase are met. Once the projects have been prioritized, the short term project must be implanted. It will mean changing attitudes, both the directors of the company and the employees, defining new roles and redesigning the company structure. The result must be that everyone in the company knows its activities, knowing what to do and how to do it. In order to ensure that the desired changes have been properly implanted, a method for quality assurance is necessary. Finally, a continuous improvement system must be designed, allowing (1) to establish in the future the medium and long term improvement projects and (2) to adjust the enterprise to the environmental changes. When these projects are completed, the integration desired will be completed.

Table 4.18.AS-IS/TO-BE analysis

Area: Macroprocess: Macroprocess: Activity[a]	Sales Order management Order processing AS-IS[b]	Document code:AS iSTobe Author: Ike J Creation date: 03-feb-2012 TO-BE[c]	File:EAsisTobe.doc Modify date:03-feb-2012 FIT[d]	 GAP + Restrictions[e]
(1.)Collect data about the order	The order is received via fax or telephone from client. If the order is received via telephone, data is written down in an orders notebook to be introduced into the IS afterwards	If the order comes via telephone. Data must be introduced straight into the IS	NOT OK	The IS does not allow real time performance due to the complexity of the consultation been carried out during the data introduction. Restriction: IS
(2.) Consult the client risk	Before introducing the order into the IS, data about client risk must be consulted. If the client risk exceeds the credit assigned, the IS does not allow the introduction of any more orders. For this reason, it is necessary to write down the order.	The order risk consultation option must be accessible from the order introduction option and must be updated with the introduction of each new detail line. The IS must allow the introduction of high risk client orders and then block them in order to avoid processing until their situation changes or a person in charge decides to block them	NOT OK	The IS does not allow the introduction of high risk and it obstructs its consultations. Restriction: IS
(3.) To introduce	Through the option		OK	

order data into the information System	"New order", the head data and detail lines of the order			

a the activity must coincide with the show activities in the textual description and the DE*/
b Describes how the activity is carried out at present

c Describes how the activity can be carried out in the future

d It must be known as follows: OK: if the activity is carried out at present in a desirable manner.: NOT OK: if the activity is not carried out according to a desirable manner

e describes the difference between the TO-BE and the AS-IS, showing possible restrictions to go from the AS-IS to the TO-BE.

D. Human Resources - Actors

As a result of the integration project, everyone in the company will know his/her activities, knowing what to do and how to do it. So it will be able to define the new company structure.

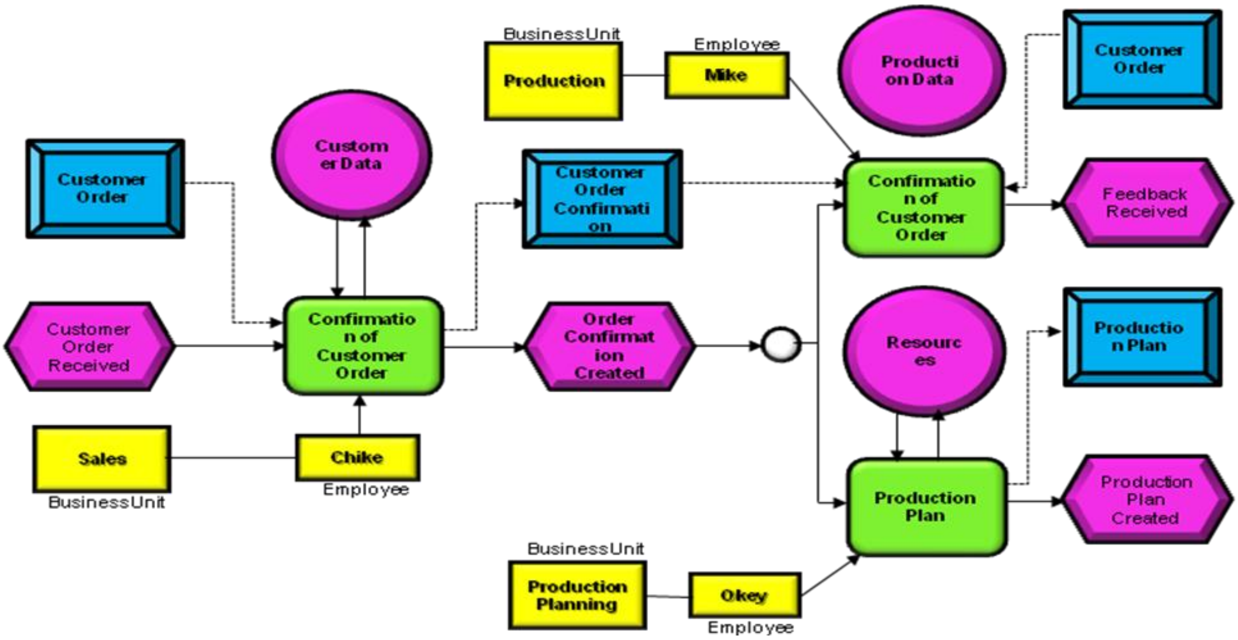


Figure 4.31 Human Resources Activities in the To-Be Ordering Process

4.6.7 Design of the Information View for the SME

The purpose of the information view is to describe the data that the SME needs to run its processes and operations. It contains data the organisation needs to support its processes. The objectives, principles and capabilities that govern this view are presented in table 4.19.

Table 4.19 Objectives, principles and capabilities of Information View

Information View	Name	Description
Objective	Define the data needed by the SME to run its processes and operations	Define the data that support business processes
Principle	Identify high level Information Models, Data organization, and Data governance	High level canonical models
Capabilities	Analysis and design of data models, logical data entity, data dictionary, ERD	Domain in the design of data models – data dictionary, ERD. Domain in logical data design

The objective of information view is to "Define the data needed by the SME to run its processes and operations with high level information model and data organisation, the capabilities of business view are "Analysis and design of data models, logical data entity, data dictionary, ERD's". Some of the current data are provided in table 4.20.

Table 4.20 Values for Instances of Information View

	Name	Description	Custodian	Domain of data	Support for business process
Domain	Account	Accounts data	Finance manager	Holds accounts information	Management of company finance
	Employee		HR manager	Stores employee information	Personnel administration
	Sales		Sales manager	Stores sales information	Management of clients
	Customer		Sales manager	Stores customer information	Management of clients
	Stock		Stock warehouse manager	Stores inventory information	Management of delivery of finished products

4.6.7.1 Identifying Data Entities for the SME

In other to develop and implement the information system, the business processes were analyzed and data entities identified. The data entities identified from the analysis are listed in table 4.21.

Table 4.21 Data Entities

SNO	Data Entity	Use
	Product Data	To store Product data
	Customer Data	To store customer data
	Financial Services Data	To store financial data
	Company Data	To store company data
	Sales Data	To store sales data
	Inventory Data	To store stock data
	Personnel Data	To store personnel data
	Purchase Data	To store purchase data
	Vendor Data	To store vendor data
	Payroll Data	To store payroll data
	Operational Data	To store daily operational data

4.6.8 Design of the Information System View for the SME

The application view contains the software products that the SME has for supporting the processes: objectives, principles and capabilities that govern this view are presented in table 4.22.

Table 4.22 Objectives, principles and capabilities of Information System View

Information System View	Name	Description
Objective	Define the best kind of applications to manage data and support business processes	Define the best application that support the business processes.
Principle	Develop Custom Integrated Application	Custom Integrated application will improve the ability to ensure ongoing maintenance and maximum value from the adoption of a packaged solution
Capabilities	Analysis, design, programming and implementation of information systems. Search packaged solutions	Domain in the analysis, design, programming and implementation of IS. Domain in search packaged solution to

	tailored to the need of the SME's	the needs of the SME. Domain to provide technical support for software and hardware

The objective of Information system view is "Define the best kinds of application to manage data and support business processes with developed custom integrated application, the capabilities for management of the information system view are "Analysis, design, programming and implementation of information systems, search packaged solution tailored to the needs of the SME's and provide technical support for software and hardware in all the company. Some of the needed applications are presented in table 4.23. In this case, the applications are related to the processes they support in the SME.

Table 4.23 Values for Instances of Information System View

	Name	**Description**	**Custodian**	**Domain of data**	**Support for business processes**
Domain	Accounts receivable and payable	Spreadsheets on Accounts	Finance manager	Update accounts information on different items	Register inputs and outputs of financial transaction
	Stock information system	Management of the inputs and outputs of the company store	Stock, warehouse manager	Update catalog of items; articles inventory processing	Register inputs and outputs of goods and raw materials
	Client IS	Manage client portfolio	Sales, sales manager	Update clients portfolio, billing	Manage client portfolio
	Human resources	Spreadsheet on personnel information	HR, Human resources manager	Update employee information. Update payroll information	Manage employees

The company process map defines the requirements of the information system which will be used to (1) develop customized software or (2) to compare the ERP software currently on the market in order to select the most appropriate one for the enterprise objectives and processes. The exhaustive requirements definition carried out according to the AIMES architecture also serves to obtain a closed contract regarding time and cost with the enterprise that develops or implements the software. Therefore, those activities for enterprise integration, according to the AIMES architecture, related to enterprise strategy analysis, enterprise processes map re-designing, improvements projects defining, etc. must be carried out before choosing whether to develop customized software or to adopt a standard ERP. In this way, an alignment between information system and enterprise strategy is achieved. In the other hand, it is shown that AIMES architecture is useful for both the development of customized systems as well as for standard ERP software implementation.

4.6.9 The Total Information Processing System (TIPS) Implementation Architecture for the SME

The Total Information Processing System (TIPS) architecture is an integrated information system implementation architecture with the objective to achieve enterprise information system integration over disparate and potentially heterogeneous systems while allowing these systems to operate independently and concurrently. A centerpiece of the integrated model is a concurrent architecture as shown is Figure 4.32 (AIMES application Architecture for TIPS). Figure 4.32 shows the architecture that describes the various components of the *information system*;

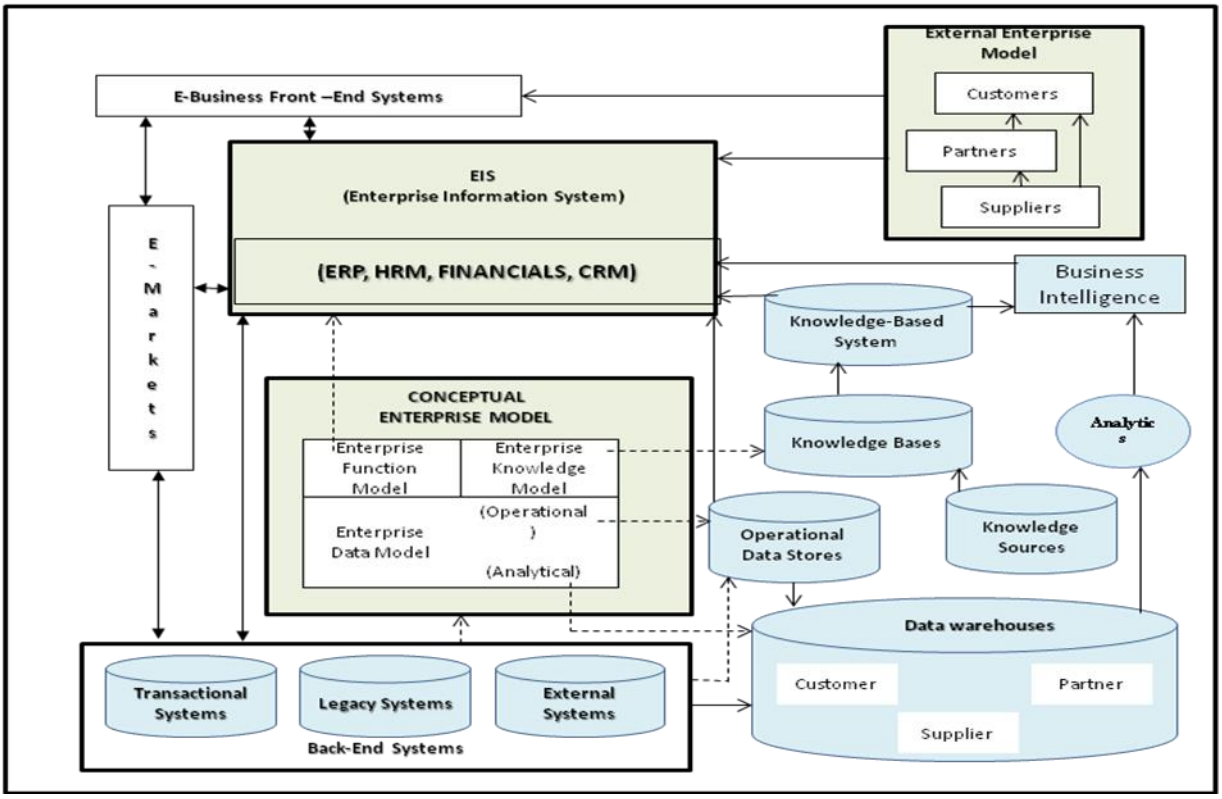


Figure 4.32 AIMES Integrated Systems Architecture (For TIPS)

4.6.9.1 Components of the Integrated System Architecture for the SME

The conceptual integrated architecture was designed as a tool to manage system complexity in the enterprise. In resolving IT system complexity using enterprise architecture, the enterprise is seen as a whole and is based on the principle of separation of concerns. This architecture is divided into five major components: the external enterprise model, the conceptual enterprise model, the front-end systems, the back-end systems and the management tools.

i). The External Enterprise Model for EIS

The external enterprise model in the architecture for EIS (Figure 4.32) consists of organizations for customers, channel partners and suppliers, the processes for operational EIS and e-marketplaces. A customer, channel partner or supplier may interact with a firm or a 3rd party e-marketplace via the e-business front-end. Operational EIS processes are the implementations of business functions described in the conceptual enterprise model. Transactions in operations are supported by the back-end transactional systems. Business intelligence from analytical processing can be used to enhance operations, whereas operational results captured by the transactional systems provide the data feeds to data warehouses for further analysis to create new business intelligence.

ii) The Conceptual Enterprise Model for EIS

The conceptual enterprise model consists of logical representations of enterprise data, knowledge and functional requirements for operations and analytics. It provides the enterprise-wide architectural blue-print for EIS. The enterprise data model consists of operational and analytic data models representing enterprise operational and analytical data requirements. The enterprise data model is the information blue-print that provides the linkage for various data structures across the enterprise. It serves as the information roadmap for system integration, new system deployment and system maintenance. The enterprise function model is the representation of business functions supporting operational and analytic functional requirements for the enterprise. It is the functional blue-print and provides the linkage within and across the external enterprise of organizational processes and the internal enterprise of computer systems. It is not uncommon to find different processes implementing the same business function to yield incompatible results. For example, a customer may receive different price quotes from different channels such as the Web, call centers or at retail outlets. Using the enterprise model framework, these processes are linked to the business function of providing price quote, which is mapped to the enterprise data model that contains the single source of pricing information (Chan, 2008).

iii). Front-End Systems

This is defined in section 4.4.1.

iv). Back-End Systems

This is defined in section 4.4.1.

v). Management Tools

Management tools are applications used to support management activities in all levels of the organisation. Management tools include applications such as *operational data stores, analytics*, data warehouse, business intelligence applications, budgeting and reporting, *Knowledge Bases and Knowledge-Based Systems*. These applications are often integrated with backend, front end and engineering applications and they collect data from these applications and refine it as useful information. Communication systems are used for collaboration and communication within the company and with external parties. Communication applications include applications such as e-mail and groupware.

4.6.9.2 Merits of the Integrated System Architecture for SME

Thus, the main **merits** of the integrated model are:

a. Real-time access to a centralized data repository containing the latest data from different applications.
b. Availability of information across platforms and departmental boundaries.
c. Rapid, secure access to vital information via software bus.
d. Easier control and management of information that is created or reused
e. Increased collaboration among employees from different departments that use different applications.
f. The insurance that returned data are always up-to-date, since all the writes and reads are made to a shared data repository
g. Data are not duplicated since they reside in their shared repository,
h. It is easier to add new sources of information by schema matching and connecting the application through the software bus.
i. The model can be easily adopted by any enterprise because of its low implementation cost.

4.7 TIPS Prototype System Design for the SME (Donavan)

UML design tools and notations are employed to graphically depict object-oriented analysis and design models. UML is a language for specifying, visualizing and constructing the artifacts of software systems, as well as for business modeling. It shows the interactions and relationships between its different classes and components. In this section we describe the TIPS system design by showing the interactions and relationships between the system's classes and components. As in any DBMS, there are multiple levels of users of the system. The following object-oriented design tools are explored to model the Total Information Processing System.

4.7.1 TIPS Use Case

The use case technique is adapted in this work to handle all objects that are needed to be present in the application as shown in Figure 4.33. It helps to determine system requirements, communication with clients and to generate test cases. From the use case diagram, the following objects to be modeled are identified: customer, personal data, sales, accounts, inventory, product, company and payroll.

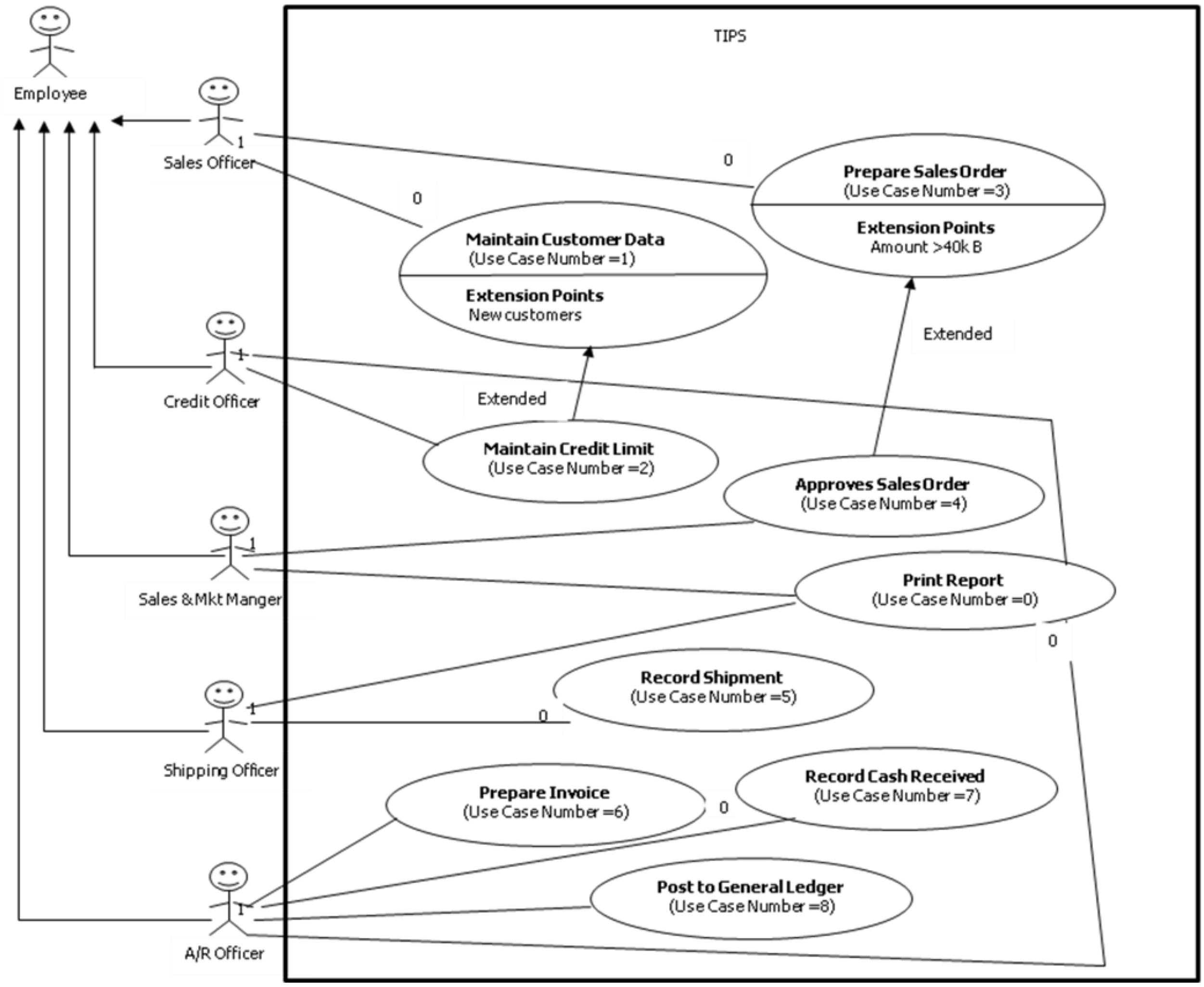
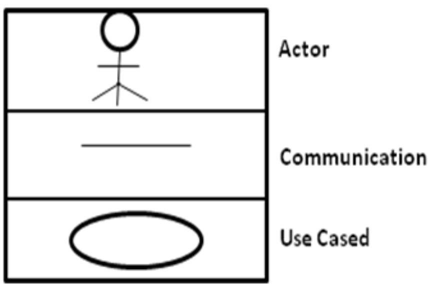


Figure 4.33 Use Case Diagram of the Proposed Prototype TIPS for the SME



4.7.2 Representation and Design of TIPS Prototype

Our analysis into the representation and design of the TIPS system starts with the process of object identification.

4.7.2.1 Object Identification in TIPS

A thorough analysis of the TIPS requirements reveals that the dominant objects within the system are: Company, Product, Customer, Financial Services, Inventory, Personnel, Vendor, Payroll, Purchase and Sales

Some of these objects exist independently, while others are compositions of other objects. Composition is a phenomenon of the object-oriented concept that allows new objects to evolve from existing objects. The objects identified are shown in Figure 4.34.

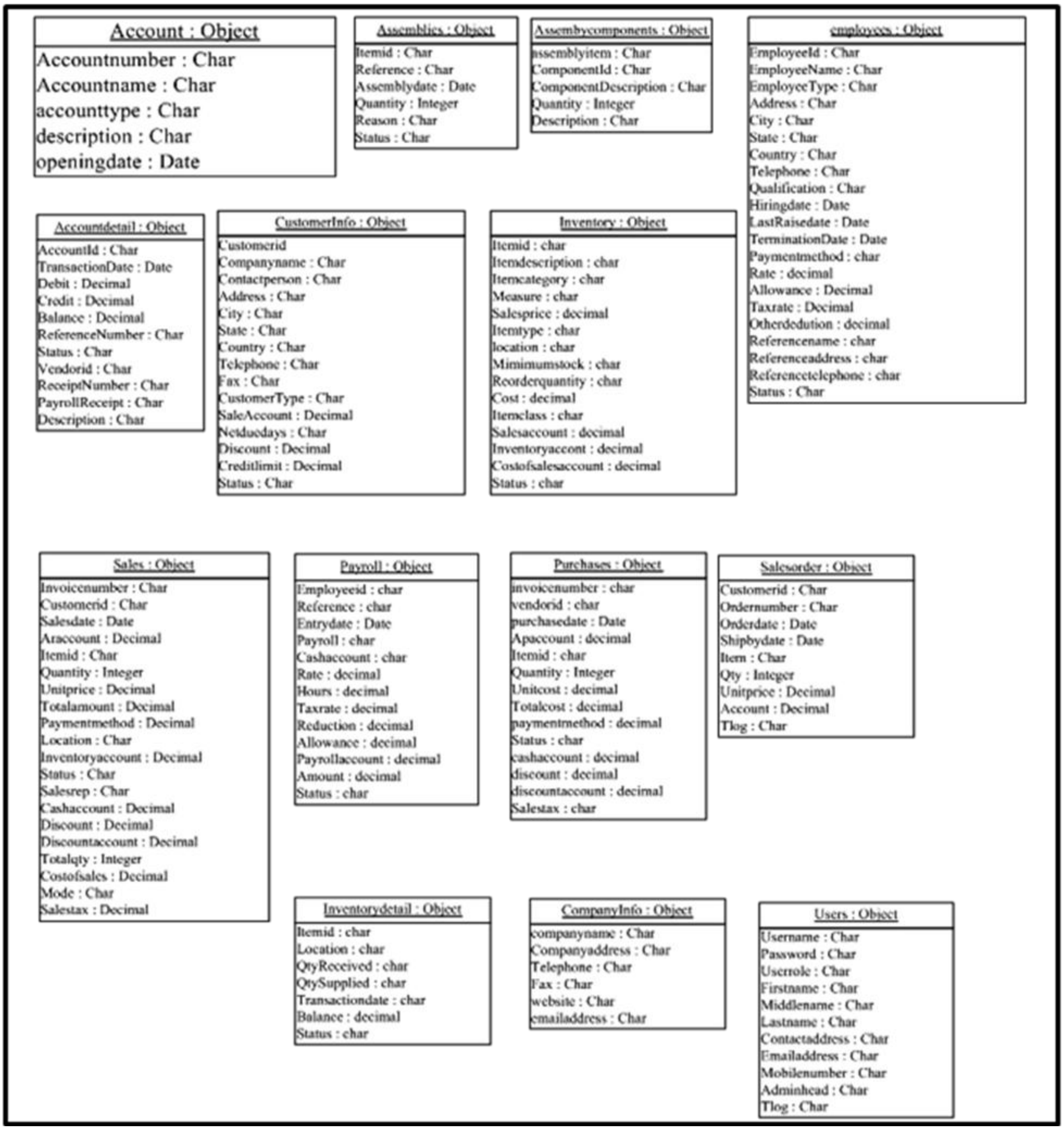


Figure 4.34 TIPS Objects

4.7.2.2 TIPS Class Diagram

TIPS class diagram in Figure 4.35 gives the overview of the system. It shows the different classes and the relationships amongst the system components. This is represented using a single class model. Our Class diagram models the Total Information Processing System. The central class is the Product; associated with it are customer, employee, sales and the reports class respectively. The UML class notation is applied with a rectangular box which is divided into three parts; name, attributes, and operation as shown in Figure 4.23.

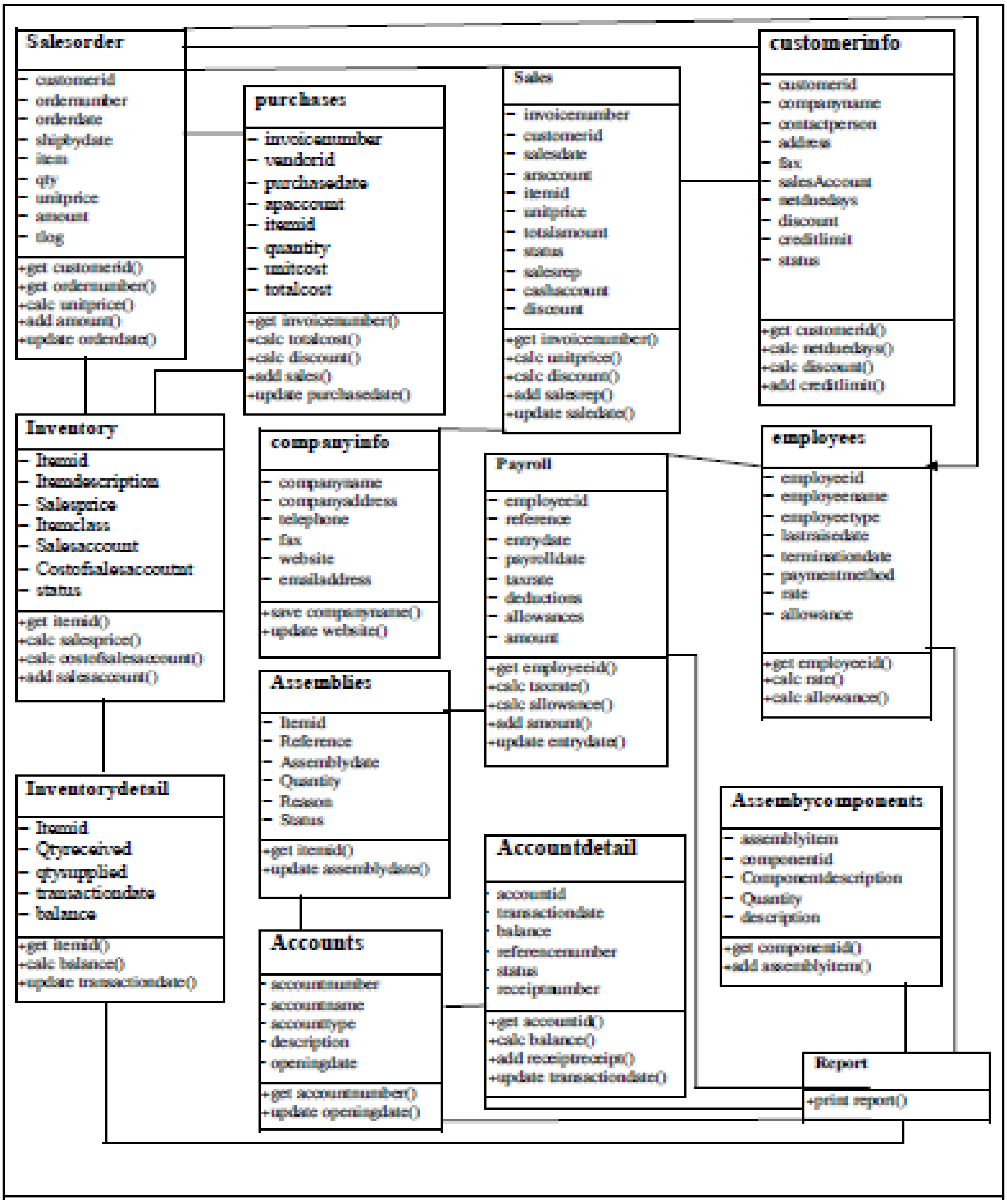


Figure 4.35 TIPS Class Diagram

4.7.3 Data Modeling

The first step in developing the new system is to build the database that will be used to store the necessary information. The database specification is defined in the data dictionary. A data dictionary contains a list of all the files in the database, the number of records in each file and the names and types of each field. The information required to be stored in the database are specified in the tables below;

Table 4.24 Accounts Table

Fieldname	Field Type	Field length	Description
Accountid	Varchar	50	The account identification number
Accountname	Varchar	50	The account name of the account
Accounttype	Varchar	50	The type of account you want to create either accounts payable, account receivable, sales, etc.
Descripti	Varchar	50	The description of the account you created
Opening date	Datetime		The opening date of the account

Table 4.25 Accounts Detail

Fieldname	Field Type	Field Length	Description
Accounted	Varchar	50	The unique account id
Transactiondate	Datetime		The transaction date for the particular transaction
Debit	Float		The debit amount
Credit	Float		The credit amount
Balance	Float		The balance of the two amount
Reference number	Varchar	50	The reference number of the transaction
Status	Varchar	50	The status of the account
Vendorid	Varchar	50	The customer identification number
Receptnumbers	Varchar	50	The auto receipt generation number holding the last receipt number
Payrollreceipt	Varchar	50	The auto receipt generation number holding the last payroll receipt number

Table 4.26 Assemblies

Field Name	Field Type	Field length	Description
Itemid	Varchar	50	The inventory item id number that you want to build
Reference	Varchar	50	The reference number of the assembly item

Assemblydate	Datetime		The date the assembly was done
Quantity	Float		The quantity of the assembly item you want
Reason	Varchar	50	The reason for the assembling
Status	Varchar	50	The status of the assembly item

Table 4.27 Assembly Components

Field Name	Field Type	Field Length	Description
Assemblyitem	Varchar	50	The assembly item you are building
Componentid	Varchar	50	The component id of the assembly item
Component description	Varchar	50	The description of the components that make up the assembly
Quantity	Float		The quantity of the assembly item to build
Status	Varchar	50	The status of the assembly item

Table 4.28 Companyinfo

Field Name	Field Type	Field Length	Description
Companyname	Varchar	100	The name of the company using the software
Companyaddress	Varchar	150	The address of the company in terms of location
Telephone	Varchar	50	The telephone number of the company
Fax	Varchar	50	The fax number if any
Website	Varchar	50	The url showing the company website address
Email	Varchar	50	The email address of the company

Table 4.29 Customerinfo

Field Name	Field Type	Field Length	Decription
Customerid	Varchar	50	The assigned customer identification name
Companyname	Varchar	100	The company name of the customer
Salesrep	Varchar	50	The contact person for the customer
Address	Varchar	150	The address of the customer
City	Varchar	50	The city of the customer
State	Varchar	20	The state of the customer
Country	Varchar	50	the country of the customer
Telephone	Varchar	50	The telephone number of the customer

Fax	Varchar	50	The fax number of the customer
Customertype	Varchar	50	The type of customer which can be cash, credit, discount etc
Salesaccount	Varchar	50	The sales account for the customer for ledger transfer
Netduedays	Float		The maximum credit day
Discount	Float		The discount percentage
Creditlimit	Float		The credit limit in terms of amount
Status	Varchar	50	The status of the customer

Table 4.30 Employees

FIELD NAME	FIELD TYPE	FIELD SIZE	FIELD DESCRIPTION
Employeeid	Varchar	10	The identification number of an employee
Employeename	Varchar	40	The employee's name
Employeetype	Varchar	40	The type of employee which can be either full time or contract
Address	Varchar	40	The address of the employee
City	Varchar	20	The city of the employee
State	Varchar	15	The state of the employee
Country	Varchar	15	the country of the employee
Telephone	Varchar	15	The telephone number of the employee
Qualification	Varchar	20	The academic qualification of the employee
Hiringdate	Datetime	Datetime	The date of employment of the employee
Lastraisedate	Datetime		The last promotion date of the employee
Terminationdate	Datetime		The appointment termination date if the appointment has been terminated
Paymentmethod	Varchar	40	The payment method of either cash or bank
Rate	Float		The hourly rate of the employee
Allowances	Float		The allowances the employee is entitled to.
Taxrate	Float		The rate of his tax
Otherdeductions	Float		Any other deduction from employees netpay
Refereename	Varchar	40	The name of employee's referee
Refereeaddress	Varchar	40	The contact address of the employee's referee
Refereetelephone	Varchar	40	The telephone of the referee
Status	Varchar	10	The status of the employee

Table 4.31 Inventory

Field Name	Field Type	Field Length	Description
Itemid	Varchar	50	The inventory item identification number
Itemdescription	Varchar	100	The item description
Itemcategory	Varchar	40	The item category either in cartons, piles etc
Measure	Float		The measure per quantity
Salesprice	Float		The item sales price
Itemtype	Varchar	50	The item type
Location	Varchar	50	The warehouse or store where the item is kept
Minimumstock	Float		The minimum stock that can be in the location before reorder
Reorderquantity	Float		The normal reorder quantity for the item
Cost	Float		The cost price of the item
Itemclass	Varchar	50	The item class which can be labour, assembly, services, stcok and non stock
Salesaccount	Varchar	50	The sales account number for ledger posting
Inventoryaccount	Varchar	50	The inventory account number for ledger posting
Costofsaleaccount	Varchar	50	The cost of sales account number for ledger posting
Status	Varchar	50	The status of the item

Table 4.32 Invetorydetail

Field Name	Field Type	Field Length	Description
Itemid	Varchar	50	The inventory item identification number
Location	Varchar	50	The warehouse or store location of the item
Qtyreceived	Float		Quantity received for the item
Qtysupplied	Float		Quantity supplied for the item
Transactiondate	Datetime		The transaction date
Balance	Float		The remaining stock
Status	Varchar	50	The status of the item

Table 4.33 Payroll

Field Name	Field Type	Field Length	Description
Employeeid	Varchar	50	The employee identification number
Reference	Varchar	50	The transaction reference number
Entrydate	Datetime		The date of entry
Payrolldate	Datetime		The payroll pay date
Cashaccount	Varchar	50	The cash account for ledger update
Rate	Float		The hourly rate for the employee
Hours	Float		The number of hours the employee worked
Taxrate	Float		The tax rate for the employee
Deductions	Float		Total deductions for the employee per month
Allowances	Float		The total monthly allowances
Payrollaccount	Varchar	50	Payroll account for ledger update
Amount	Float		The gross amount for the month
Status	Varchar	50	The status of the payment

Table 4.34 Purchases

Field Name	Field Type	Field Length	Description
Invoivenumber	Varchar	50	The invoice number
Vendorid	Varchar	50	The vendor identification number
Purchasedate	Datetime		The date of purchase
Apaccount	Varchar	50	The ledger account code
Itemid	Varchar	50	The item code purchased
Quantity	Float		The quantity of the item purchased
Unitcost	Float		The unit cost for each item
Totalcost	Float		The total cost of the item quantity
Paymethod	Varchar	50	The payment method either cash, card etc.
Location	Varchar	50	The location of the item in store
Inventoryaccount	Varchar	50	The ledger inventory account to inventory
Status	Varchar	50	The status of the item transaction
Cashaccount	Varchar	50	The ledger account to ledger
Discount	Float		The discount percentage given to the vendor
Discountaccount	Varchar	50	The ledger discount account to be

			updated
Salestax	Float		The vat percentage per sales

Table 4.35 Sales

Field Name	**Field Type**	**Field Length**	**Description**
Invoicenumber	Varchar	50	The invoice number for the transaction
Customerid	Varchar	50	The customer identification number
Saledate	Datetime		The date of sales
Araccount	Varchar	50	The account receivable ledger account
Itemid	Varchar	50	The item code
Quantity	Float		The item quantity purchased
Unitprice	Float		The unit price of the item
Totalamount	Float		The total amount for the quantity purchased.
Paymethod	Varchar	50	The payment method for the transaction either cash, credit card or cheque.
Location	Varchar	50	The item location in the warehouse
Inventoryaccount	Varchar	50	The ledger account for inventory to be updated.
Status	Varchar	50	The status of the sales
Salesrep	Varchar	50	The sales representative for the customer for commission.
Cashaccount	Varchar	50	The ledger account for cash to be updated
Discount	Float		The discount given customer
Discountaccount	Varchar	50	The ledger account for discount to be updated
Totalqty	Float		The total quantity purchased
Costofsales	Float		The total cost of the sales
Mode	Varchar	30	The mode of transaction
Salestax	Float		The vat percentage

Table 4.36 Salesorder

Field Name	**Field Type**	**Field Length**	**Description**
Customerid	Varchar	50	The customer identification number
Ordernumber	Varchar	50	The order number for easy tracing
Orderdate	Datetime		The date of order
Shipbydate	Datetime		The ship date of the product
Item	Varchar	150	The item ordered
Qty	Float		The quantity of item ordered

Unitprice	Float		The unit price of the item
Amount	Float		The total amount of order
Tlog	Varchar	100	The audit log file

Table 4.37 Users

Field Name	**Field Type**	**Field Length**	**Description**
Username	Varchar	40	The user identification for the system
Password	Varchar	50	The password of the user
Userrole	Varchar	30	The role of the user in the software
Firstname	Varchar	40	The users firstname
Middlename	Varchar	40	Users middlename
Lastname	Varchar	40	Last name of the user
Contactaddress	Varchar	150	The user's contact address
Emailaddress	Varchar	100	The email address
Mobilenumber	Varchar	30	The mobile phone
Adminhead	Varchar	30	The manger of the user
Tlog	Varchar	100	The audit log

4.7.3.1 The Entity Relationship Data (ERD) model

An entity-relationship (E-R) data model is a high-level conceptual model that describes data as entities, attributes, identifiers and relationships.

Entity Class: This is the structural description of things that share common attributes

Attribute: Describes an entity class

Identifier: Identifies an entity instance. A unique value for each entity instance.

Relationship: The interactions among entities

Figure 4.36 shows the ERD for TIPS database.

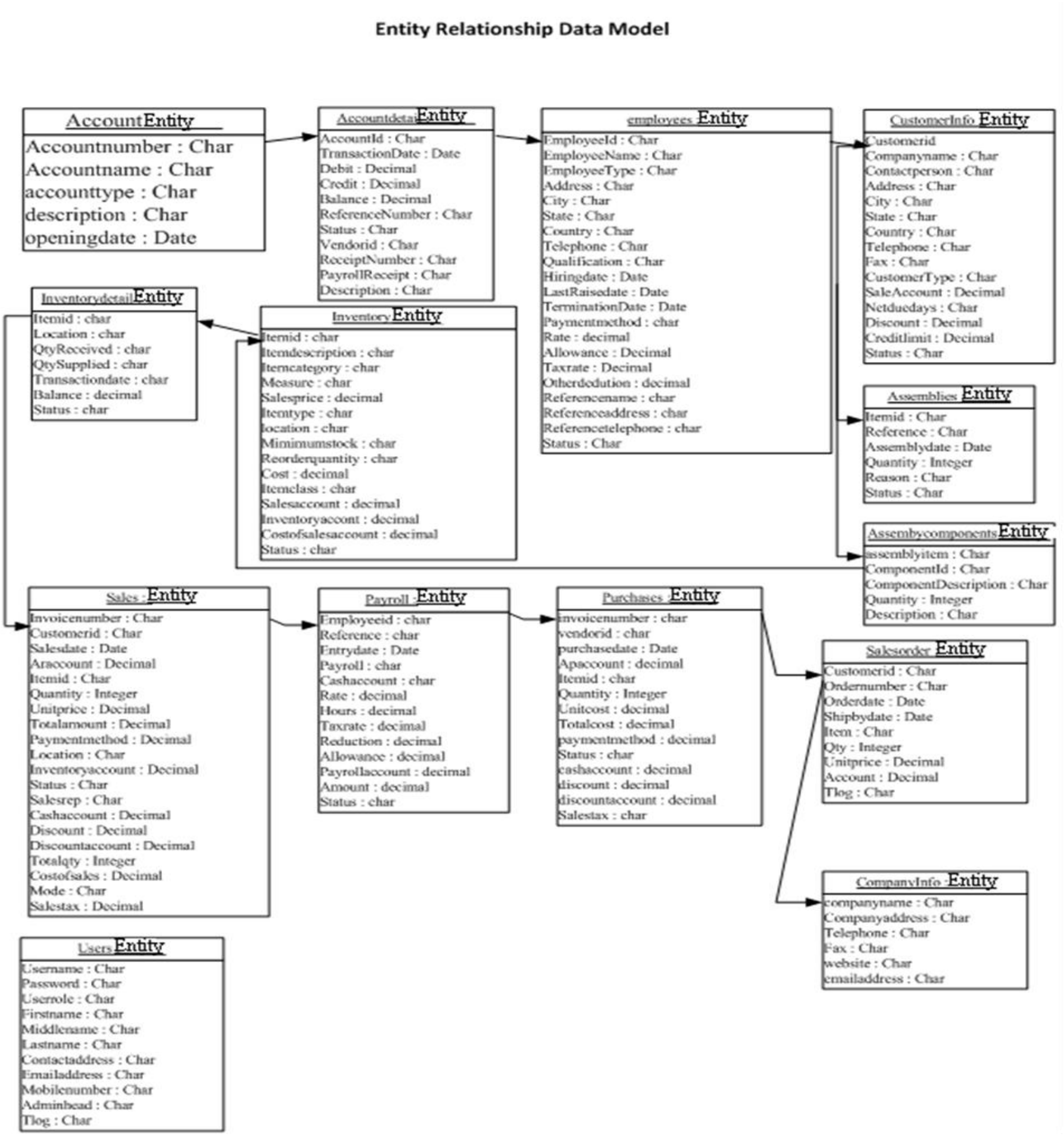


Figure 4.36 Entity Relationship Diagram

4.7.3.2 Data Dictionary

The data dictionaries are tables that contain the descriptions of classes and the types, descriptions, and constraints on attributes of an information system.

Table 4.38 Entity: Accounts

S/N	FIELDNAME	DESCRIPTION	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	Accountnumber	The account identification number	Character	50		Yes	Yes	pk
2	accountname	The account name of the account	character	50		Yes	No	
3	accounttype	The type of account you want to create	character	50		Yes	No	

		either accounts payable, account receivable, sales, etc.						
4	description	The description of the account you created	character	100		Yes	No	
5	openingdate	The opening date of the account	datetime .	8		yes	No	

Table 4.39 Entity: Accountdetail

S/N	FIELDNAME	DESCRIPTION	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	accountid	The unique account id	Character	50		yes	Yes	PK
2	transactiondate	The transaction date for the particular transaction	Datetime	8		Yes	No	
3	Debit	The debit amount	Numeric	8		Yes	No	
4	Credit	The credit amount	Numeric	8		Yes	No	
5	Balance	The balance of the two amount	Numeric	8		yes	No	
6	Referencenumber	The reference number of the transaction	Character	50		Yes	Yes	
7	Status	The status of the account	Character	50		Yes	No	
8	Vendorid	The customer identification number	Character	50		Yes	Yes	FK
9	receiptnumber	The auto receipt generation number holding the last receipt number	Character	50		Yes	Yes	
10	payrollreceipt	The auto receipt generation number holding the last payroll receipt number	Character	50		Yes	No	
11	description	Description of the account type	Character	150		no	No	

Table 4.40 Entity: customerinfo – This table contains information about different customers

S/N	FIELDNAME	DESCRIPTION	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	customerid	The assigned customer identification name	Character	50		Yes	Yes	PK
2	Companyname	The company name of the customer	Character	100		Yes	No	
3	contactperson	The contact person for the customer	Character	100		Yes	No	
4	Address	The address of the customer	Character	100		No	No	
5	City	The city of the customer	Character	40		No	No	
6	State	The state of the customer	Character	40		No	No	
7	Country	the country of the customer	Character	40		No	No	
8	telephone	The telephone number of the customer	Character	40		No	No	
9	Fax	The fax number of the customer						
9	customertype	The type of customer which can be cash, credit, discount etc.	Character	50		No	No	
10	salesAccount	The sales account for the customer for ledger transfer	Character	50		Yes	No	
11	netduedays	The maximum credit day	Numeric	8		No	No	
12	Discount	The discount percentage	Numeric	8		No	No	
13	creditlimit	The credit limit in terms of amount	Numeric	8		Yes	No	
14	Status	The status of the customer	Character	50		No	No	

Table 4.41 Entity: Assemblies– This table contains information about different assembly combination

S/N	FIELDNAME	DESCRIPTION	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	Itemid	The inventory	Characte	50		yes	yes	PK

		item id number that you want to build	r					
2	Reference	The reference number of the assembly item	character	50		yes	No	
3	Assemblydat e	The date the assembly was done	datetime	8		yes	No	
4	Quantity	The quantity of the assembly item you want	numeric	8		no	No	
5	Reason	The reason for the assembling	character	100		no	No	
6	Status	The status of the assembly item	character	50		no	No	

Table 4.42 Entity: Assembycomponents – This table contains information about different assembly combination

S/ N	FIELDNAME	DESCRIPTI ON	TYPE	FIELD WIDT H	DEFAUL T	REQUIRE D	UNIQU E	KEY(S)
1	assemblyitem	The assembly item you are building	characte r	50		yes	No	
2	componentid	The component id of the assembly item	characte r	50		yes	Yes	
3	Componentdescript ion	The description of the components that make up the assembly	Charact er	50		no	No	
4	Quantity	The quantity of the assembly item to build	numeric	8		yes	No	
5	description	The status of the assembly item	Charact er	50		no	No	

Table 4.43 Entity: companyinfo– This table contains information about the company

S/ N	FIELDNAME	DESCRIPTIO N	TYPE	FIELD WIDTH	DEFA ULT	REQUIRE D	UNIQU E	KEY(S)
1	companyname	The name of the company using the software	character	100		yes	No	
2	Companyaddres s	The address of the company in terms of location	character	150		no	No	
3	telephone	The telephone	character	50		no	no	

		number of the company						
4	Fax	The fax number if any	character	50		no	No	
5	Website	The url showing the company website address	character	50		no	No	
6	emailaddress	The email address of the company	character	50		no	No	

Table 4.44 Entity: employees – This table contains information about the company employee

S/N	FIELDNAME	DESCRIPTION	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	employeeid	The identification number of an employee	character	50		Yes	Yes	PK
2	employeename	The employee's name	character	80		Yes	No	
3	employeetype	The type of employee which can be either full time or contract	character	40		Yes	No	
4	Address	The address of the employee	character	150		Yes	No	
5	City	The city of the employee	character	40		No	No	
6	State	The state of the employee	character	40		No	No	
7	Country	the country of the employee	character	40		No	No	
8	telephone	The telephone number of the employee	character	40		No	No	
9	qualification	The academic qualification of the employee	character	50		Yes	No	
10	hiringdate	The date of employment of the employee	datetime	8		Yes	No	
11	lastraisedate	The last promotion date of the employee	datetime	8		Yes	No	
12	terminationdate	The appointment termination date if the appointment has been terminated	datetime	8		No	No	

13	paymentmethod	The payment method of either cash or bank	character	50		Yes	No	
14	Rate	The hourly rate of the employee	numeric	8		Yes	No	
15	allowance	The allowances the employee is entitled to.	numeric	8		Yes	No	
16	Taxrate	The rate of his tax	numeric	8		Yes	No	
17	otherdeductions	Any other deduction from employees netpay	numeric	8		Yes	No	
18	refereename	The name of employee's referee	character	80		Yes	No	
19	refereeaddress	The contact address of the employee's referee	character	150		Yes	No	
20	Refereetelephone	The telephone of the referee	character	50		Yes	No	
21	Status	The status of the employee	character	50			No	

Table 4.45 Entity: Inventory – This table contains information about the stock of the company

S/N	FIELDNAME	DESCRIPTION	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	Itemid	The inventory item identification number	character	50		Yes	Yes	PK
2	Itemdescription	The item description	character	100		Yes	No	
3	Itemcategory	The item category either in cartons, piles etc	character	40		No	No	
4	Measure	The measure per quantity	Numeric	8		No	No	
5	Salesprice	The item sales price	Numeric	8		No	No	
6	Itemtype	The item type	character	50		No	No	
7	Location	The warehouse or store where the item is kept	character	50		Yes	No	
8	Minimumstock	The minimum stock that can be in the location before reorder	Numeric	8		Yes	No	

9	Reorderquantity	The normal reorder quantity for the item	Numeric	8		Yes	No	
10	Cost	The cost price of the item	Numeric	8		Yes	No	
11	Itemclass	The item class which can be labour, assembly, services, stcok and non stock	character	50		No	No	
12	Salesaccount	The sales account number for ledger posting	character	50		No	No	
13	Inventoryaccount	The inventory account number for ledger posting	character	50		No	No	
14	Costofsalesaccoutnt	The cost of sales account number for ledger posting	character	50		No	No	
15	Status	The status of the item	character	50		No	No	

Table 4.46 Entity: Inventorydetail – This table contains information about the stock of the company

S/N	FIELDNAM E	DESCRIPTIO N	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	Itemid	The inventory item identification number	character	50		Yes	Yes	PK
2	Location	The warehouse or store location of the item	characterr	50		Yes	No	
3	Qtyreceived	Quantity received for the item	numeric	8		No	No	
4	qtysupplied	Quantity supplied for the item	numeric	8		No	No	
5	Transactionda te	The transaction date	datetime	8		No	No	
6	balance	The remaining stock	numeric	8		No	No	
7	Status	The status of the item	character	50		Yes	No	

Table 4.47 Entity: Payroll – This table contains information about the stock of the company

S/ N	FIELDNAME	DESCRIPTIO N	TYPE	FIELD WIDT H	DEFAUL T	REQUIRE D	UNIQU E	KEY(S)

1	employeeid	The employee identification number	character	50		yes	Yes	PK
2	reference	The transaction reference number	character	50		yes	Yes	
3	entrydate	The date of entry	datetime	8		yes	No	
4	payrolldate	The payroll pay date	datetime	8		yes	No	
5	cashaccount	The cash account for ledger update	character	50		yes	No	
6	Rate	The hourly rate for the employee	numeric	8		yes	No	
7	Hours	The number of hours the employee worked	numeric	8		yes	No	
8	Taxrate	The tax rate for the employee	numeric	8		yes	No	
9	deductions	Total deductions for the employee per month	numeric	8		yes	No	
10	allowances	The total monthly allowances	numeric	8		yes	No	
11	Payrollaccount	Payroll account for ledger update	character	50		yes	No	
12	Amount	The gross amount for the month	numeric	8		yes	No	
13	Status	The status of the payment	character	50		yes	No	

Table 4.48 Entity: Purchases – This table contains information about purchases from the company

S/N	FIELDNAME	DESCRIPTION	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	Invoicenumber	The invoice number	character	50		yes	Yes	PK
2	Vendorid	The vendor identification number	character	50		yes	Yes	FK
3	Purchasedate	The date of purchase	datetime	8		yes	No	
4	Apaccount	The ledger account code	character	50		yes	No	
5	Itemid	The item code purchased	character	50		yes	Yes	
6	Quantity	The quantity of the item purchased	numeric	8		yes	No	
7	unitcost	The unit cost	numeric	8		yes	No	

		for each item						
8	Totalcost	The total cost of the item quantity	numeric	8		yes	No	
9	Paymentmethod	The payment method either cash, card etc.	character	50		yes	No	
10	Status	The location of the item in store	character	50		yes	No	
11	Cashaccount	The ledger inventory account to inventory	character	50		yes	No	
12	Discount	The status of the item transaction	numeric	8		yes	No	
13	Discountaccount	The ledger account to ledger	Character	50		yes	No	
14	Salestax	The discount percentage given to the vendor	numeric	8		yes	No	

Table 4.49 Entity: sales – This table contains information about sales by the company

S/N	FIELDNAME	DESCRIPTION	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	Invoicenumber	The invoice number for the transaction	character	50		yes	Yes	PK
2	Customerid	The customer identification number	character	50		yes	Yes	FK
3	Salesdate	The date of sales	datetime	8		yes	No	
4	Araccount	The account receivable ledger account	character	50		yes	No	
5	Itemid	The item code	character	50		yes	Yes	
6	Quantity	The item quantity purchased	numeric	8		yes	No	
7	Unitprice	The unit price of the item	numeric	8		yes	No	
8	Totalamount	The total amount for the quantity purchased.	numeric	8		yes	No	
9	Paymentmethod	The payment method for the transaction either cash, credit card or cheque.	character	50		yes	No	

10	Location	The item location in the warehouse	character	50		yes	No	
11	Inventoryaccount	The ledger account for inventory to be updated.	character	50		yes	No	
12	Status	The status of the sales	character	20		yes	No	
13	Salesrep	The sales representative for the customer for commission.	character	50		yes	No	
14	Cashaccount	The ledger account for cash to be updated	character	50		yes	No	
15	Discount	The discount given customer	numeric	8		yes	No	
16	Discountaccount	The ledger account for discount to be updated	character	50		yes	No	
17	Totalqty	The total quantity purchased	numeric	8		yes	No	
18	Costofsales	The total cost of the sales	numeric	8		yes	No	
19	Mode	The mode of transaction	character	40		yes	No	
20	Salestax	The vat percentage	numeric	8		yes	No	

Table 4.50 Entity: salesorder – This table contains information about order from the company

S/N	FIELDNAME	DESCRIPTION	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	customerid	The customer identification number	character	50		yes	Yes	PK
2	ordernumber	The order number for easy tracing	character	50		yes	Yes	FK
3	orderdate	The date of order	datetime	8		yes	No	
4	shipbydate	The ship date of the product	datetime	8		yes	No	
5	Item	The item ordered	character	50		yes	Yes	
6	Qty	The quantity of item ordered	numeric	8		yes	No	
7	Unitprice	The unit price of the item	numeric	8		yes	No	
8	Amount	The total amount of	numeric	8		yes	No	

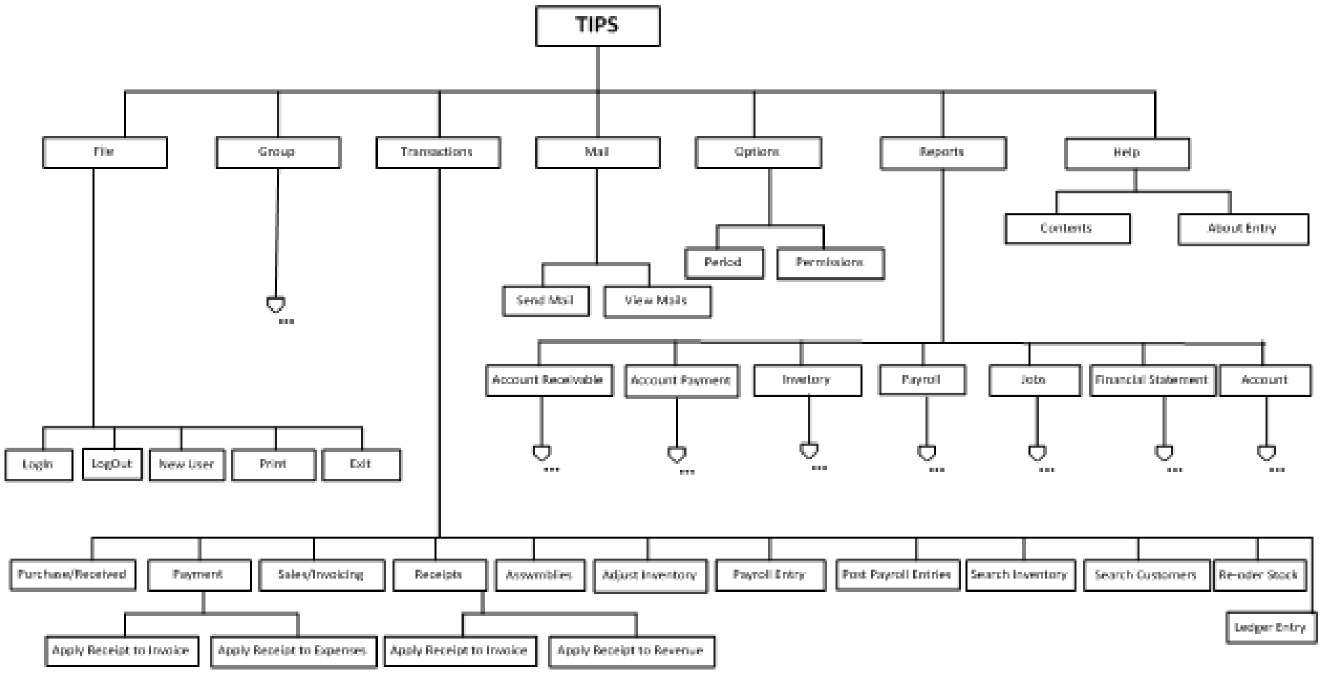
		order						
9	Tlog	The transaction audit log file	character	50		yes	No	

Table 4.51 Entity: users – This table contains information about users of the information system

S/N	FIELDNAME	DESCRIPTION	TYPE	FIELD WIDTH	DEFAULT	REQUIRED	UNIQUE	KEY(S)
1	username	The user identification for the system	character	40		yes	Yes	PK
2	password	The password of the user	character	50		yes	Yes	
3	Userrole	The role of the user in the softeware	character	30		yes	No	
4	firstname	The users firstname	character	40		No	No	
5	middlename	Users middlename	Character	40		No	No	
6	lastname	Last name of the user	character	40		No	No	
7	Contactaddress	The user's contact address	character	150		No	No	
8	emailaddress	The email address	character	50		No	No	
9	mobilenumber	The mobile phone	character	30		No	No	
10	adminhead	The manger of the user	character	30		No	No	
11	Tlog	The transaction audit log	character	100		yes	No	

4.7.4 Object Aggregation in TIPS

Object aggregation is generally used to illustrate the static structure of an object-oriented system. It shows the details of how different objects are „part of" other objects. This makes it possible to identify objects that can be represented as sub-objects of other objects. The object aggregation in TIPS is shown in Figure 4.37.



N.B: The aggregation is shown using links annotated with circular blob meaning 'Part of'. The off page reference suggest that further details about objects are not shown

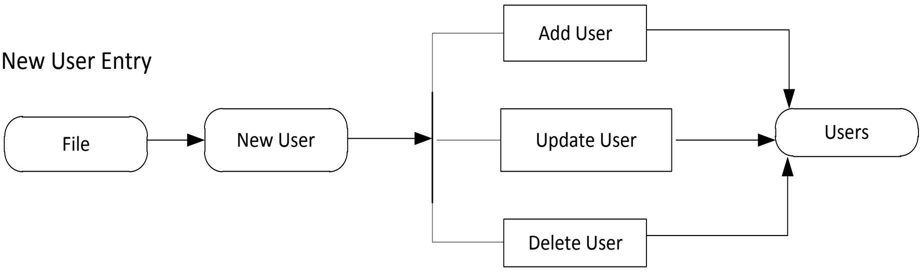
Figure 4.37 Object Aggregation in TIPS

4.7.4.1 Object Interaction in TIPS

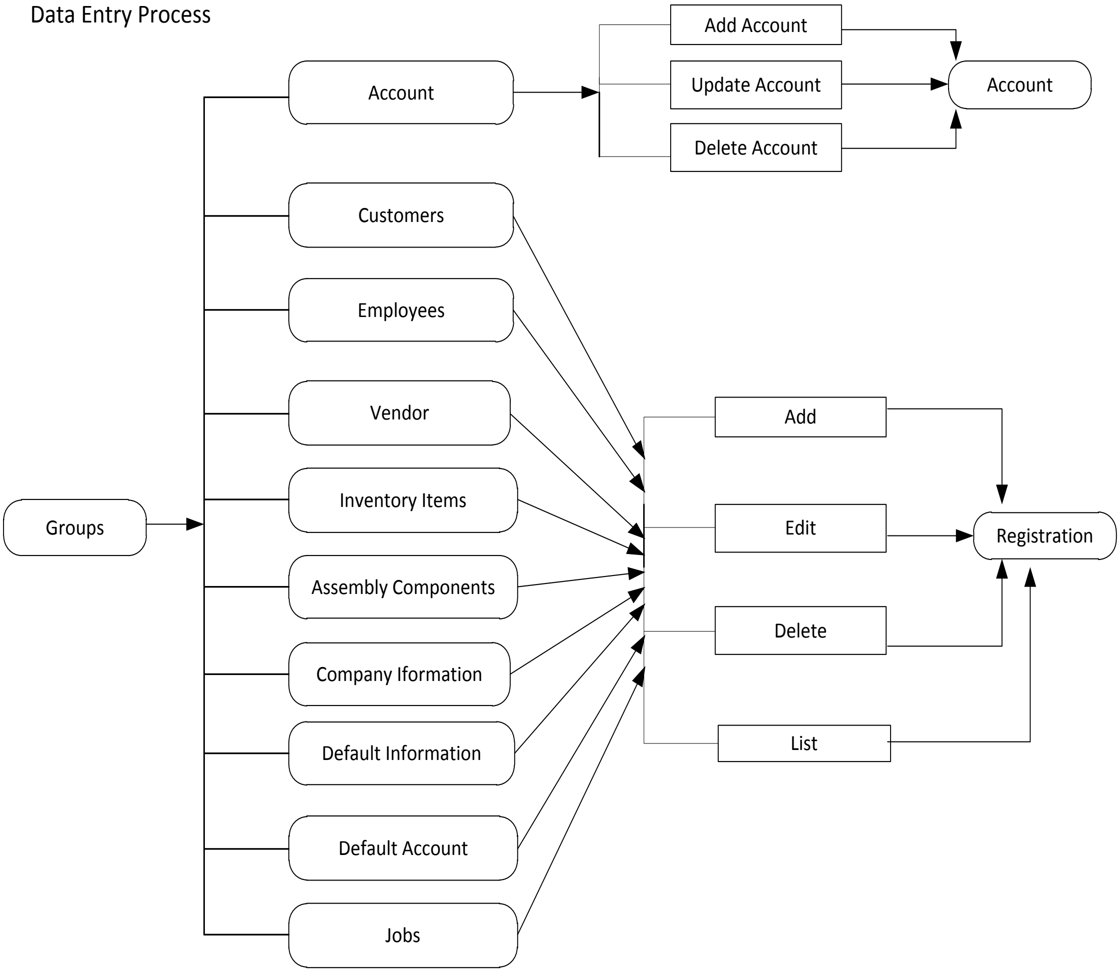
The Object interaction phase of the design process reveals the dynamic structure of object to object interaction within the system when the system is executing. In order words, it shows how objects interact with one another, how service calls and requests are passed between different objects. Figure 4.38 shows the object interaction among some of the objects in TIPS.

OBJECTS INTERACTIONS

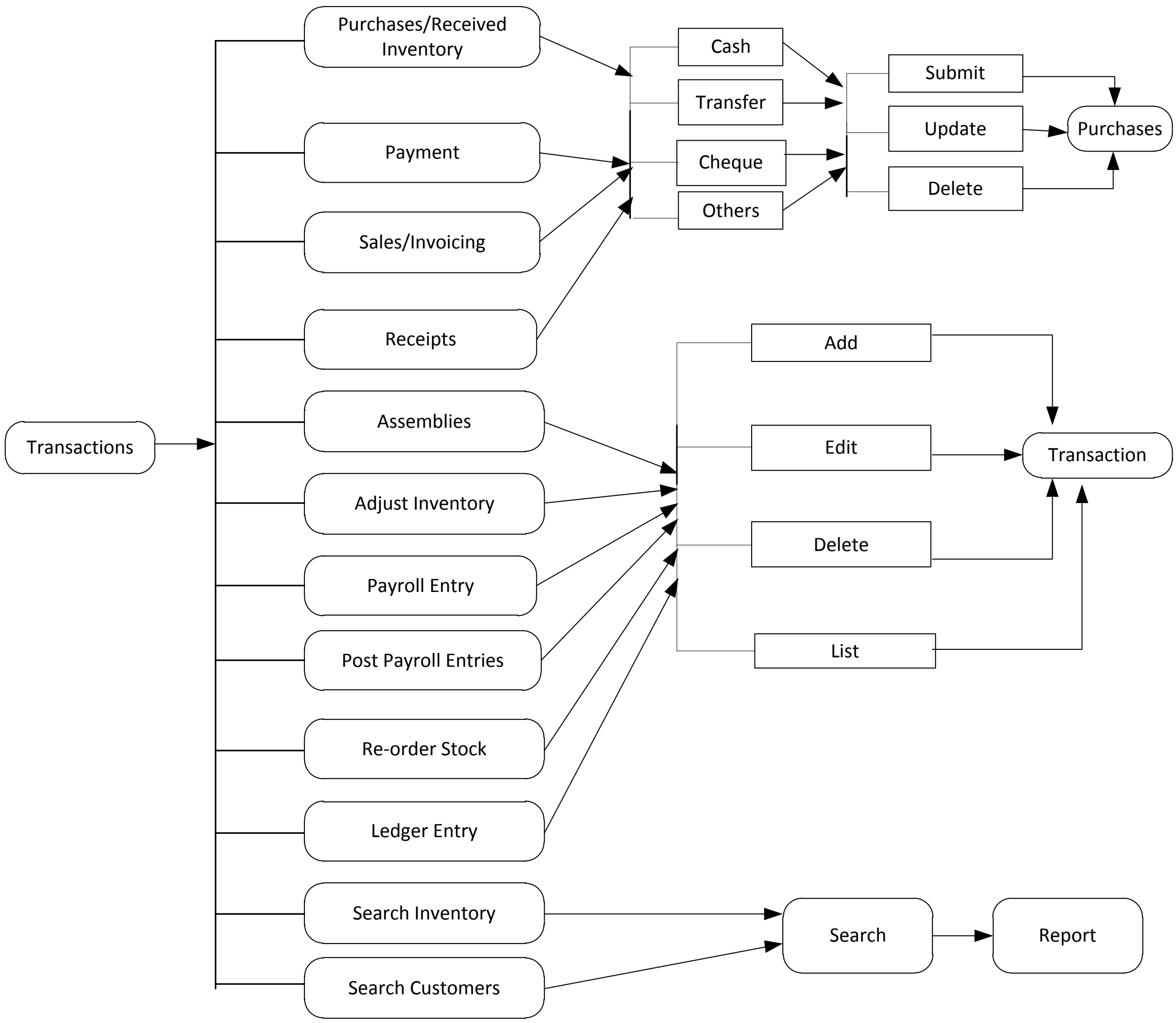
New User Entry



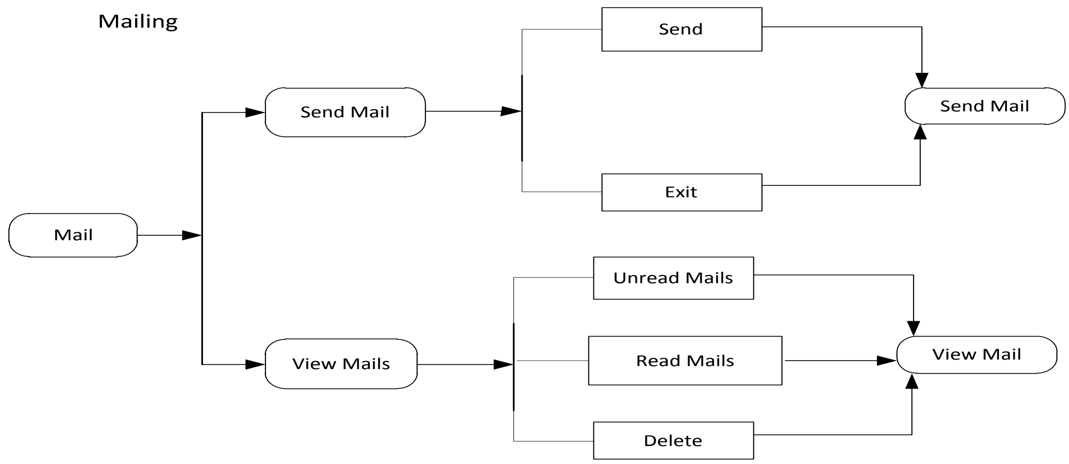
Data Entry Process



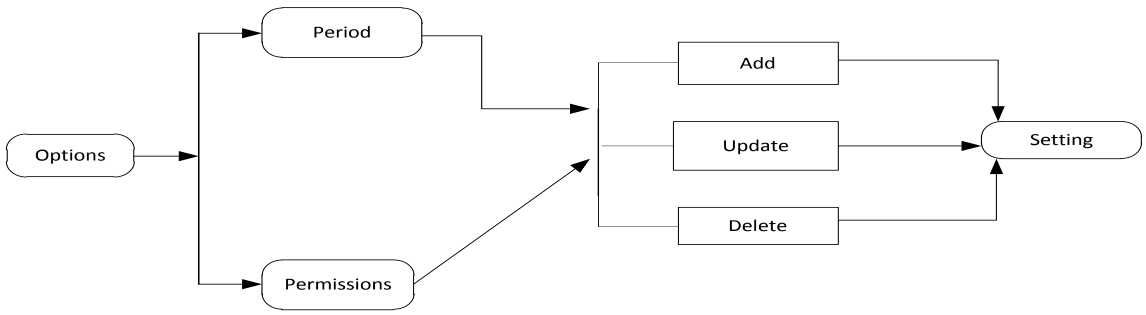
Transactions Process



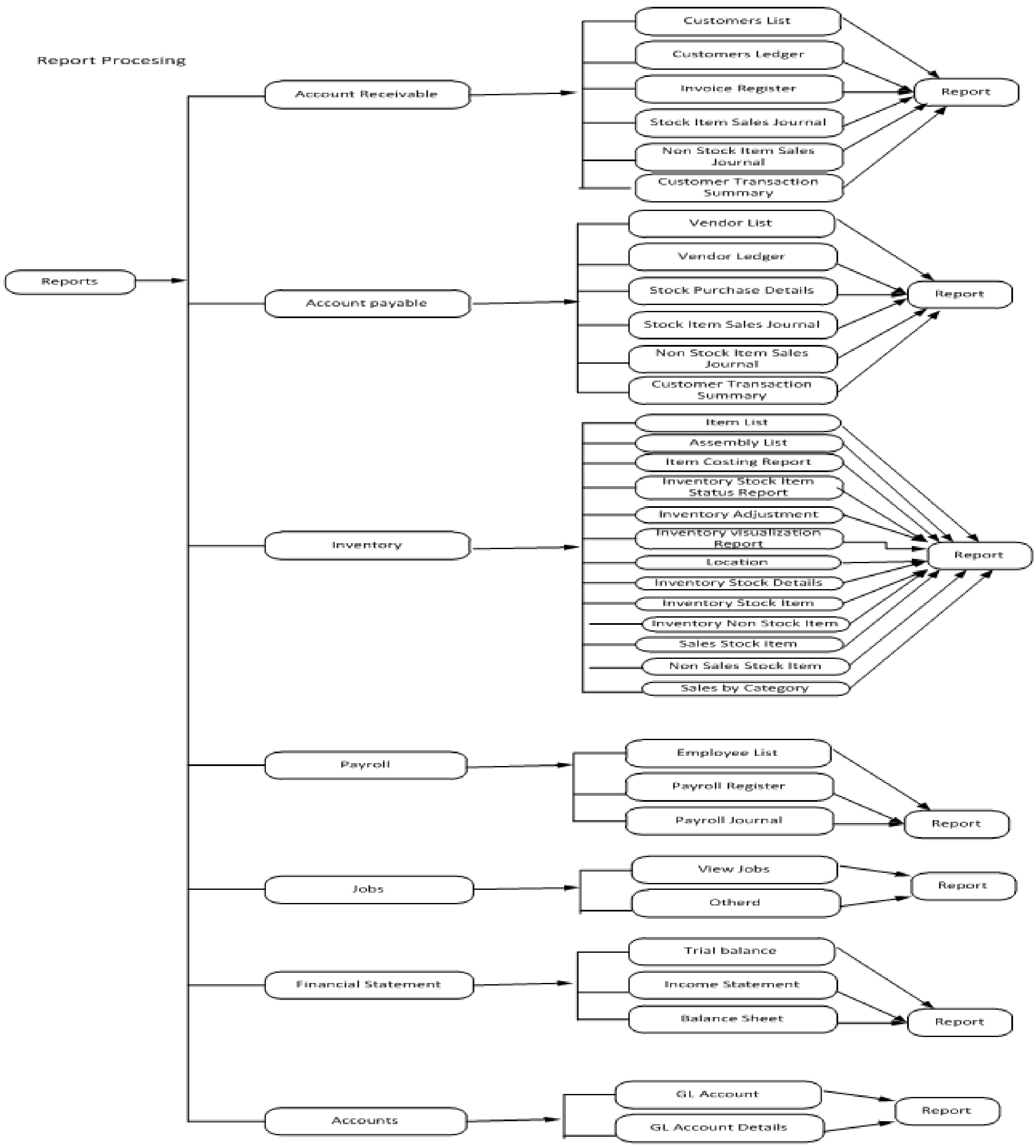
Mailing

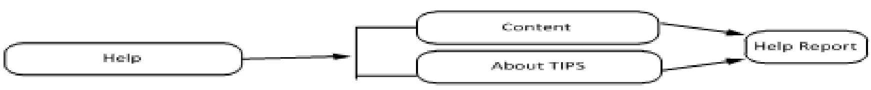


Settings



Report Procesing





4.7.4.2 Object Interface Design for TIPS

Having represented, in details, the static and dynamic structure of objects within the TIPS, we now show the specification of the object design interfaces. This involves defining the types of the object attributes, the signatures and semantics of the objection operations as shown in figure 4.39.

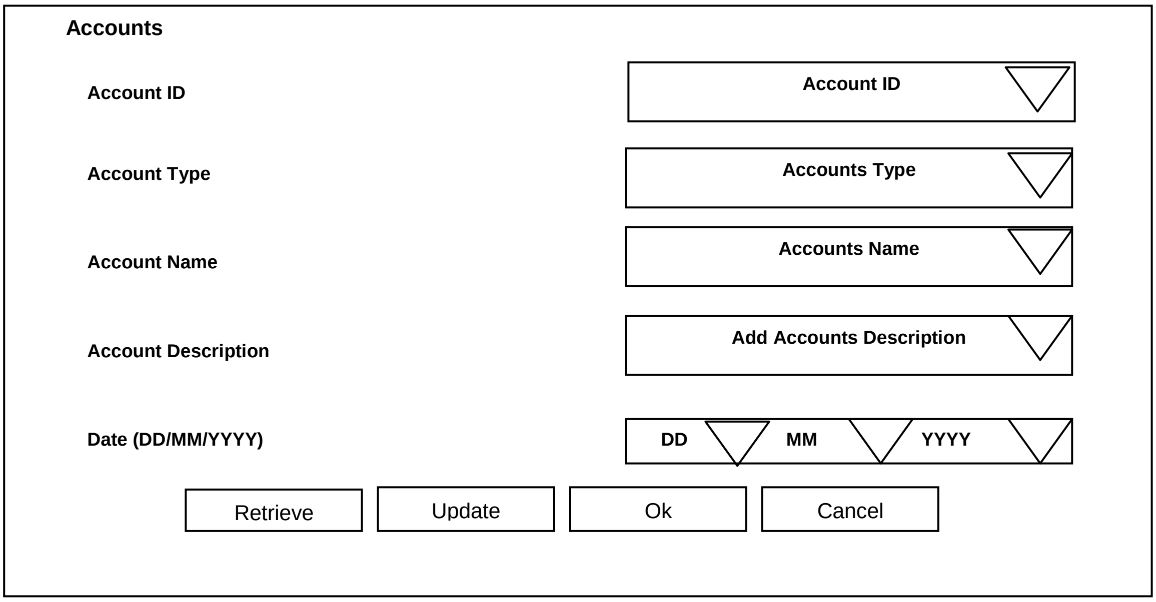


Figure 4.39 Interface for Accounts

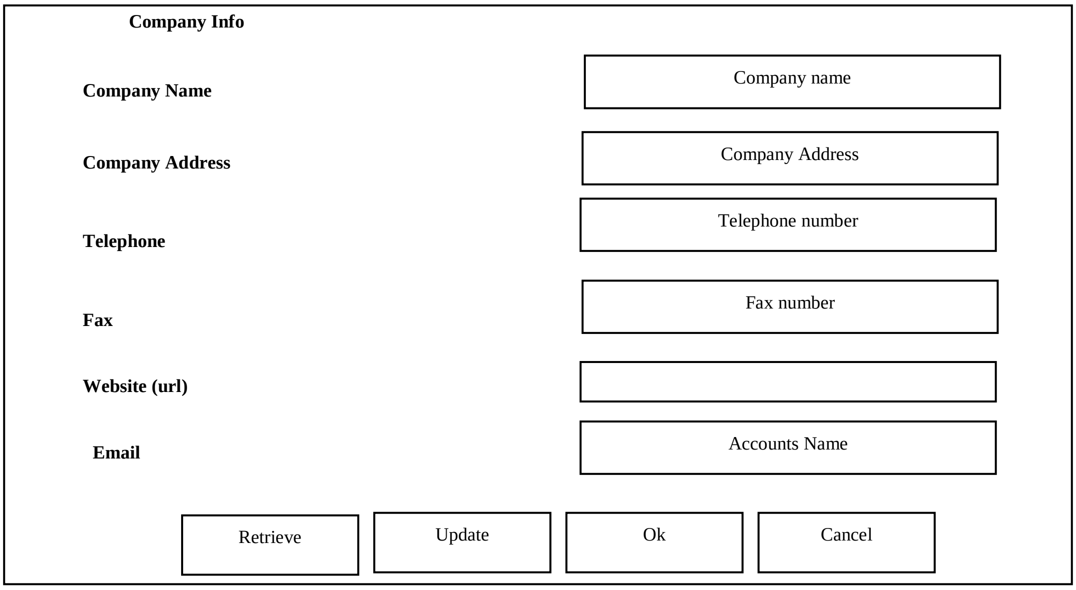


Figure 4.40 Interface for Company Information

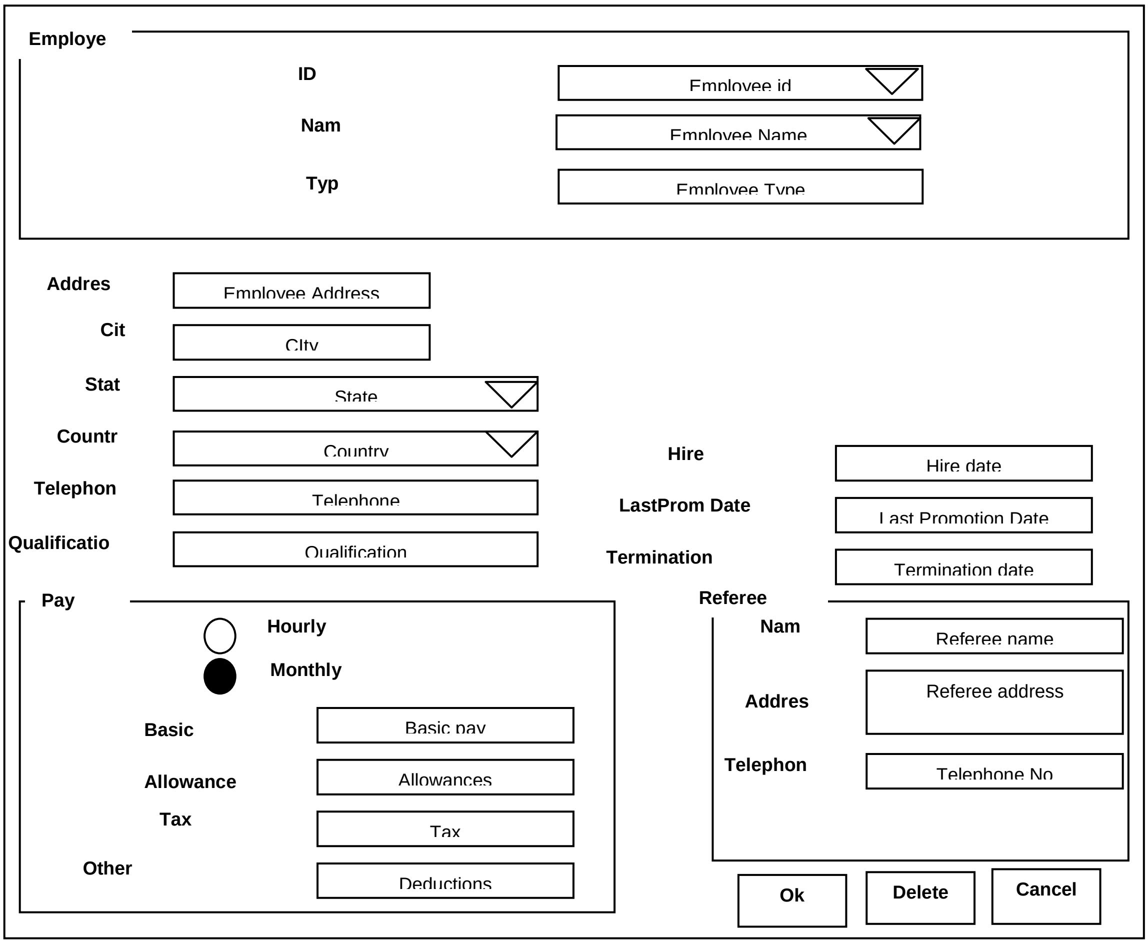


Figure 4.41 Interface for Employee

4.7.5 System Architecture for the SME

To implement the application, client-server architecture is required. The most popular client-server architectures are the two-tier and the three-tier architecture. The choice of architecture affects the development time and the future flexibility and maintenance of the application. While selecting the architecture most suitable for an application, many factors including the complexity of the application, the number of users and their geographical dispersion are considered. This system is designed based on a traditional three-tier architecture used by many client-server applications. Three-tier architecture includes a presentation layer, business rules/ logic layer, and the data layer. The three-tier architecture is shown in Figure 4.42.

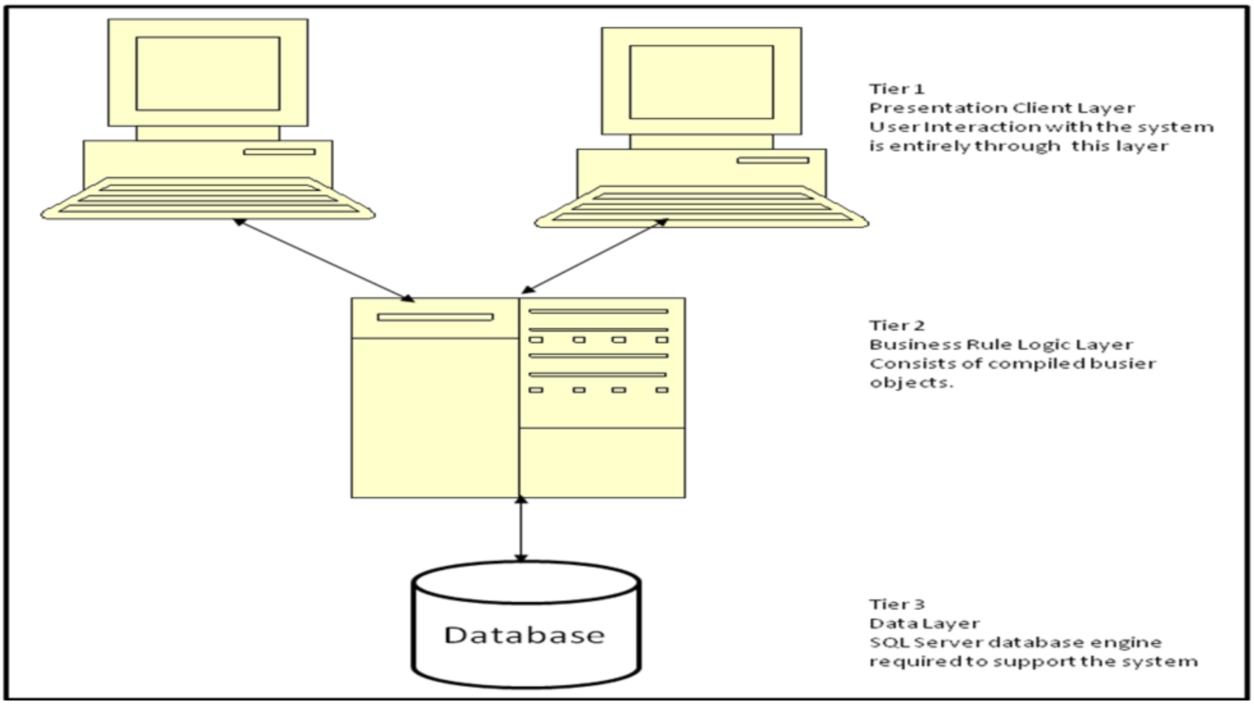


Figure 4.42 Three tier architecture for the SME

Data Base Tier 1: Presentation/ Client Layer user interaction with the system is entirely through this layer. Tier 2: Business Rules/Logic Layer consists of compiled business objects, components and classes Tier 3: Data Layer SQL Server or any other database engine required to support web application. Figure 4.42, Three-tier architecture is generally used when an effective distributed client/server design is needed that provides · increased performance · flexibility · maintainability · reusability and · scalability. This model hides the complexity of distributed processing from the user. These features have made the three-tier architecture a popular choice over the two-tier architecture for Internet applications. The three layers are discussed below. The Data layer is responsible for data storage. Primarily this tier (layer) consists of one or more relational databases and/or file systems. The Business Rules/Logic layer is the middleman between the presentation layer and the data layer. This middle tier was introduced to overcome the deployment limitation (whenever the application logic changed the application had to be redistributed at each and every client) in the two-tier architecture. The middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users. The Presentation Layer, also called the Client tier, is responsible for the presentation of data, receiving user events, and controlling the user interface. The user interaction with the system is entirely through this layer.

4.8. Program Modules

The program module specifies the various modules that will run the system. These modules are specified in Table 4.52.

Table 4.52 Program Modules

s/n	Module Name	Description
1	DBStuff. Bas	This module handles most database activity. When you run the application for the first time, it will create all the required tables, queries, views needed by the application in the database
2	MailStuff.bas	This module handles the mail part of the application. Helps in sending and receiving mails between users of the application.
3	MainRoutine.bas	This is the main routine module that loads all the needed necessary forms and classes required for the system to work efficiently and effectively.
4	RegTableStuff.bas	This module registers the application in the registry of the machine (computer system) once you run it for the first time on the machine. It edits the Hkey Local machine and adds the application title and configuration
5	UserModule.bas	This module handles most of the activities that has to do with user permissions and roles.
6	Mail	This module handles the internal messaging system of the enterprise taking care of receiving and sending mail between clients

The output Format

The system generates reports for all level of management starting from Strategic to Tactic and to operations. Some of the reports include:

Financial Statements (Trial Balance, Income Statement, Balance Sheet)

Payroll (Employee List, Payroll Register, Payroll Journal)

Account Receivables (Customer List, Invoice Register)

And so many other reports that will help the management carry out their responsibility of control, supervision and Coordination.

4.9 Overall Data Flow Diagram of The System

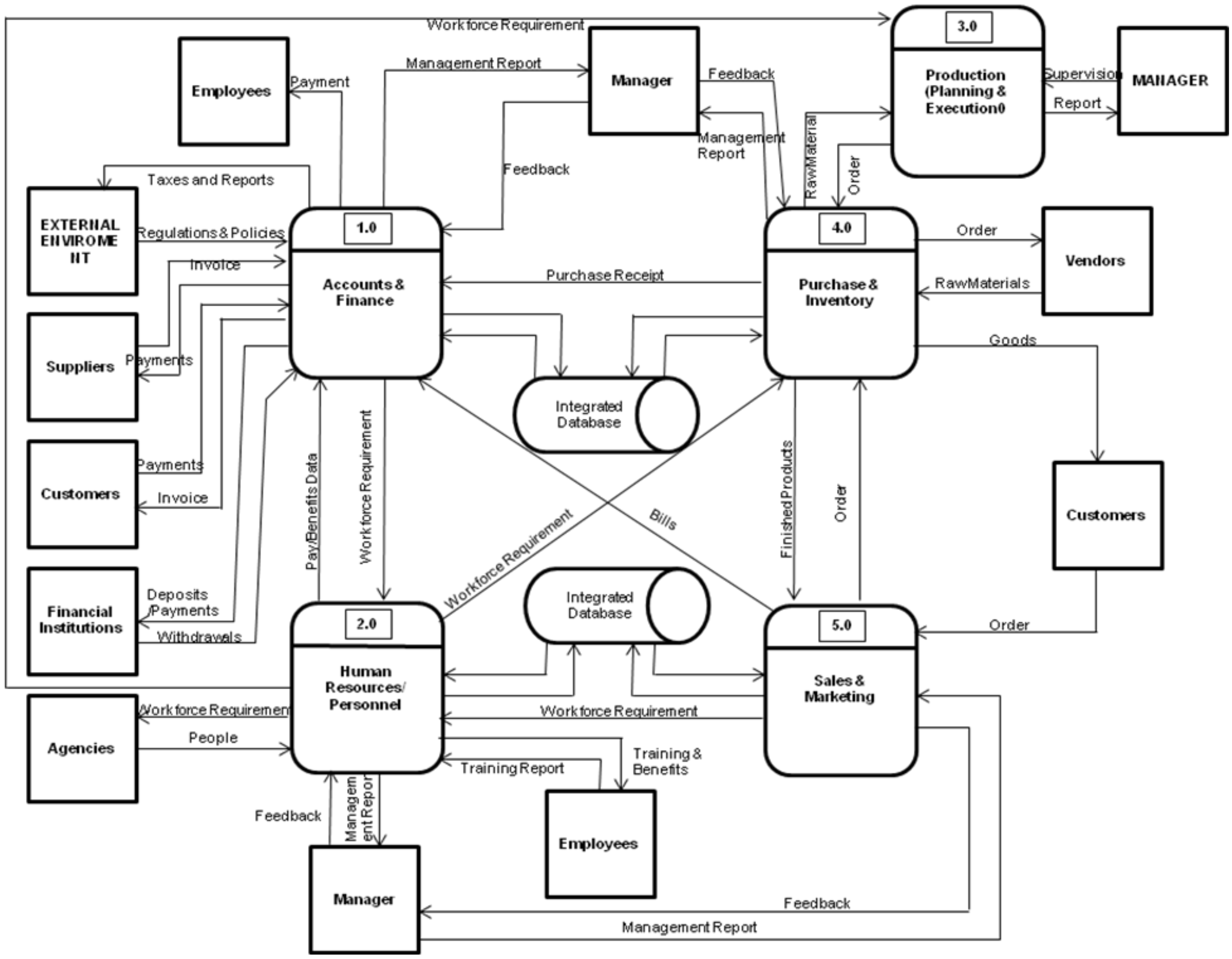


Figure 4.43 Overall Data Flow Diagram of the System

4.10. Flowcharts of the Proposed TIPS

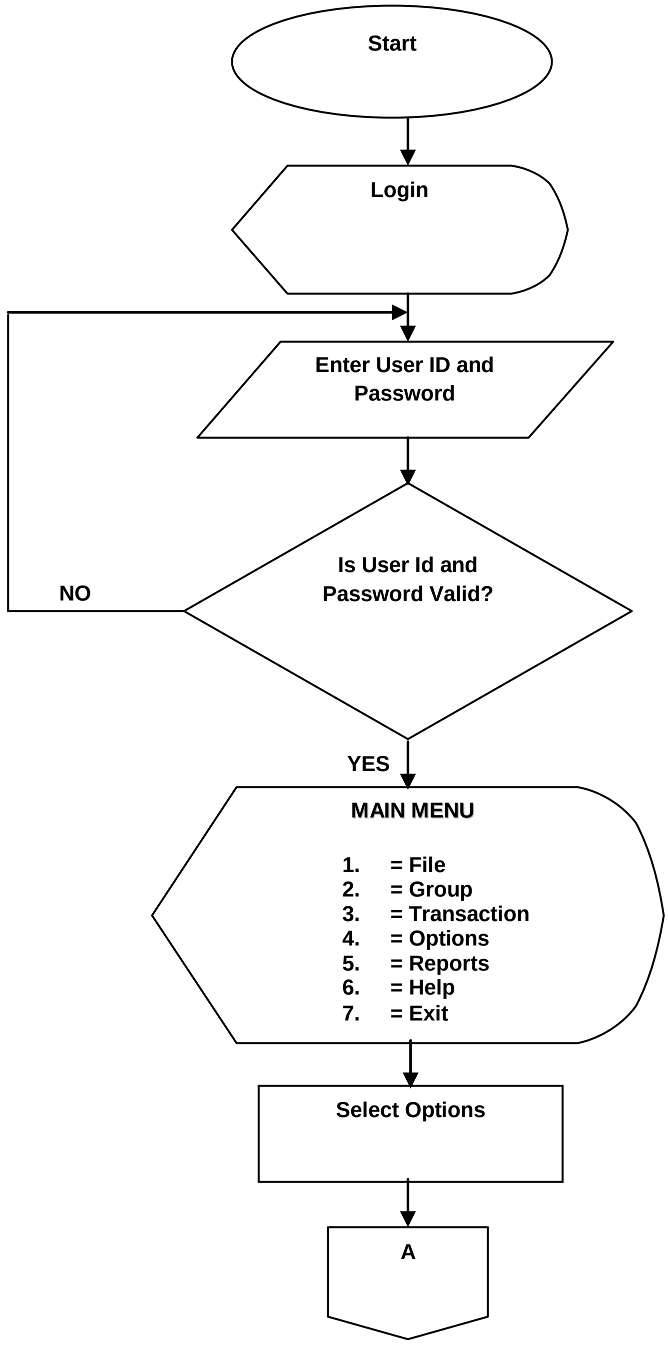


Figure 4.44 Flowchart of the Proposed TIPS

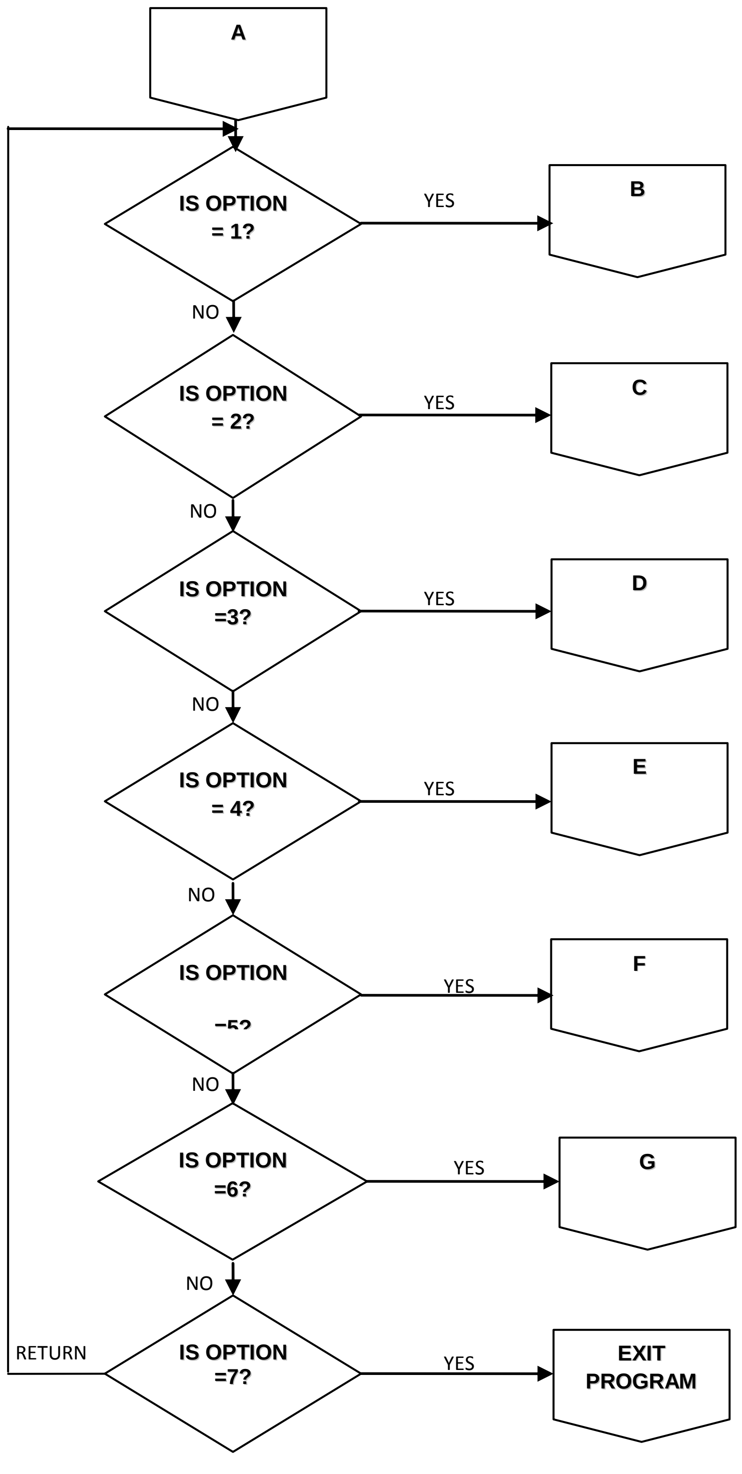


Figure 4.45 Main Menu flowchart

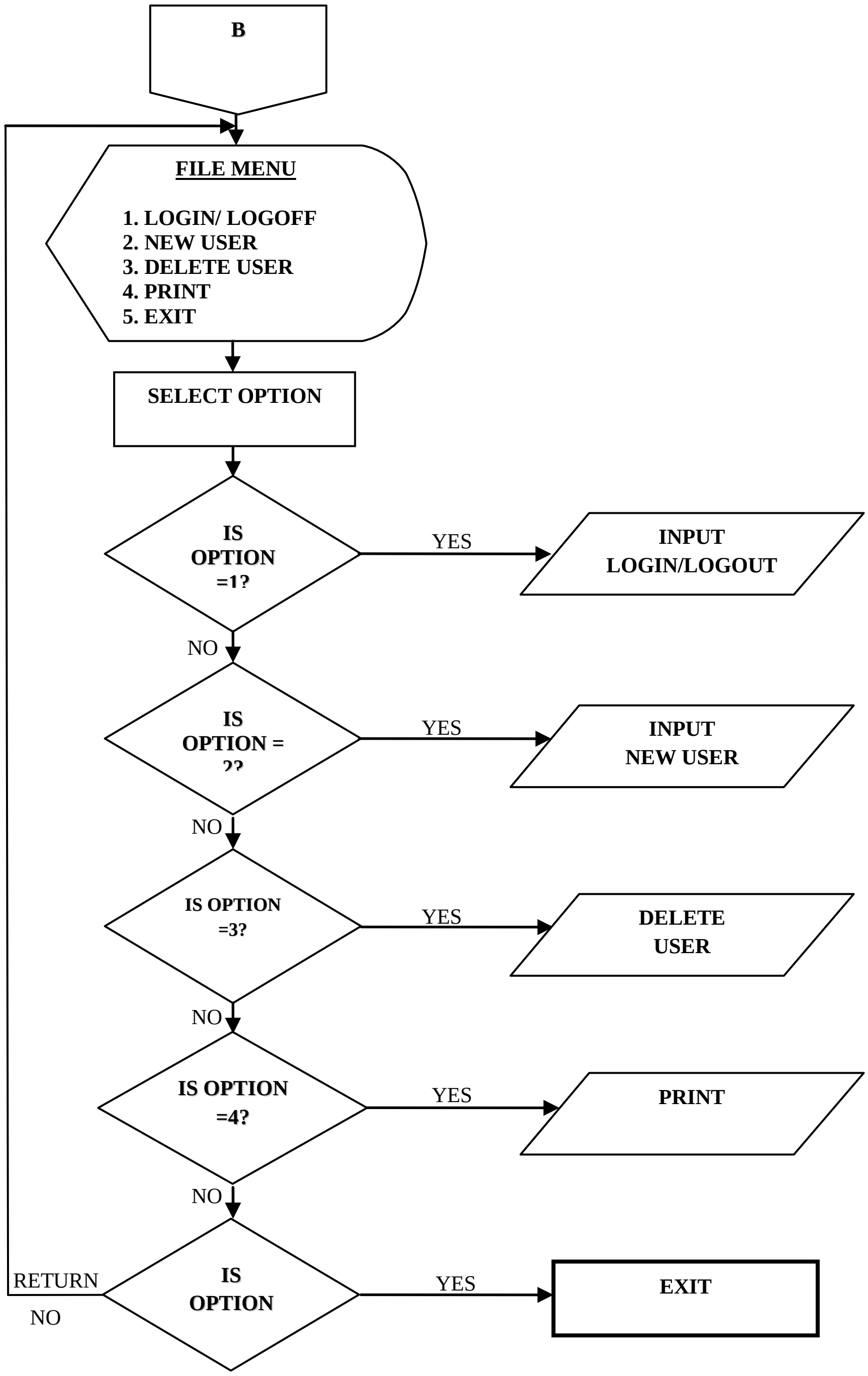


Figure 4.46 File Subsystem flowchart

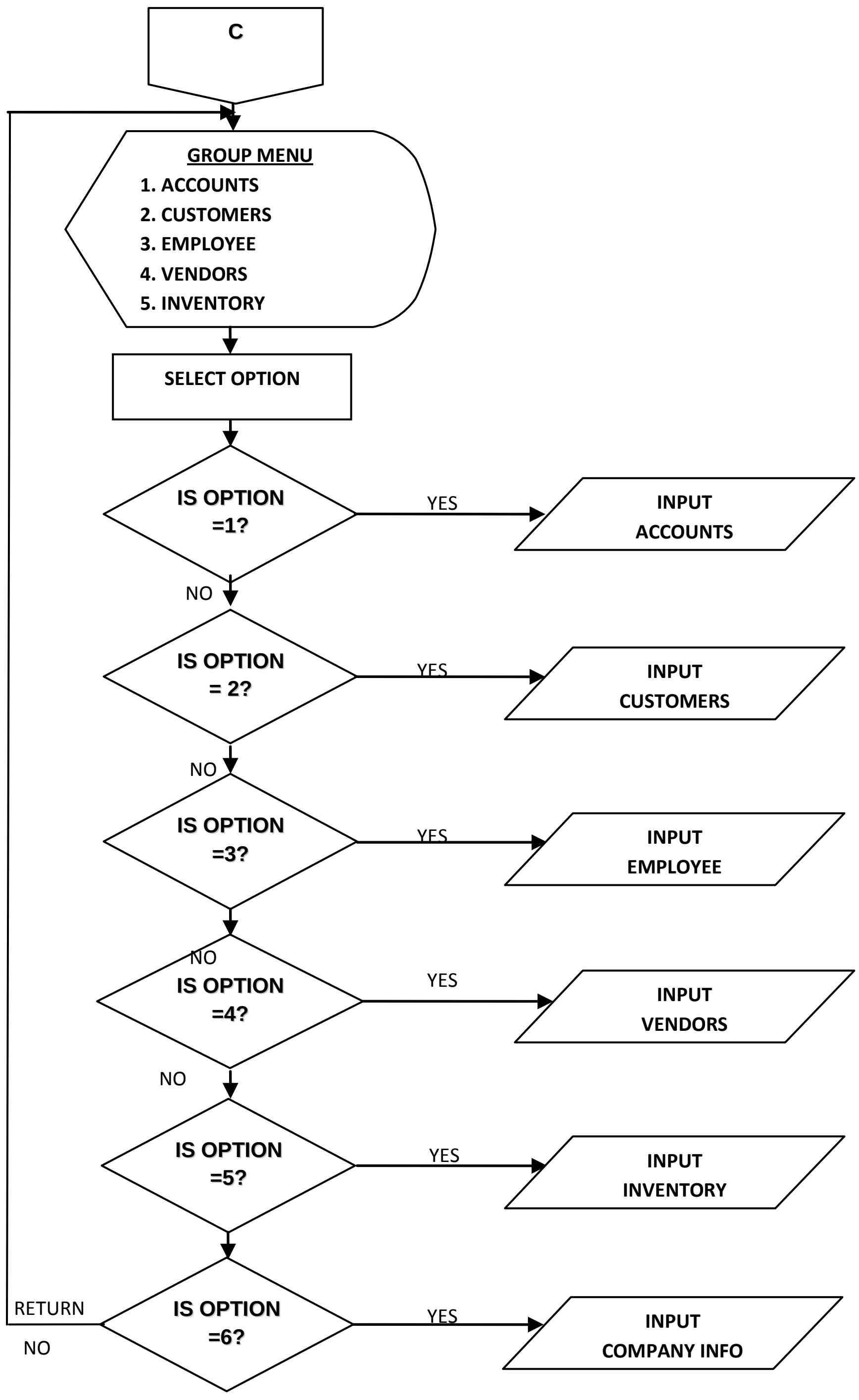


Figure 4.47 Group Subsystem flowchart

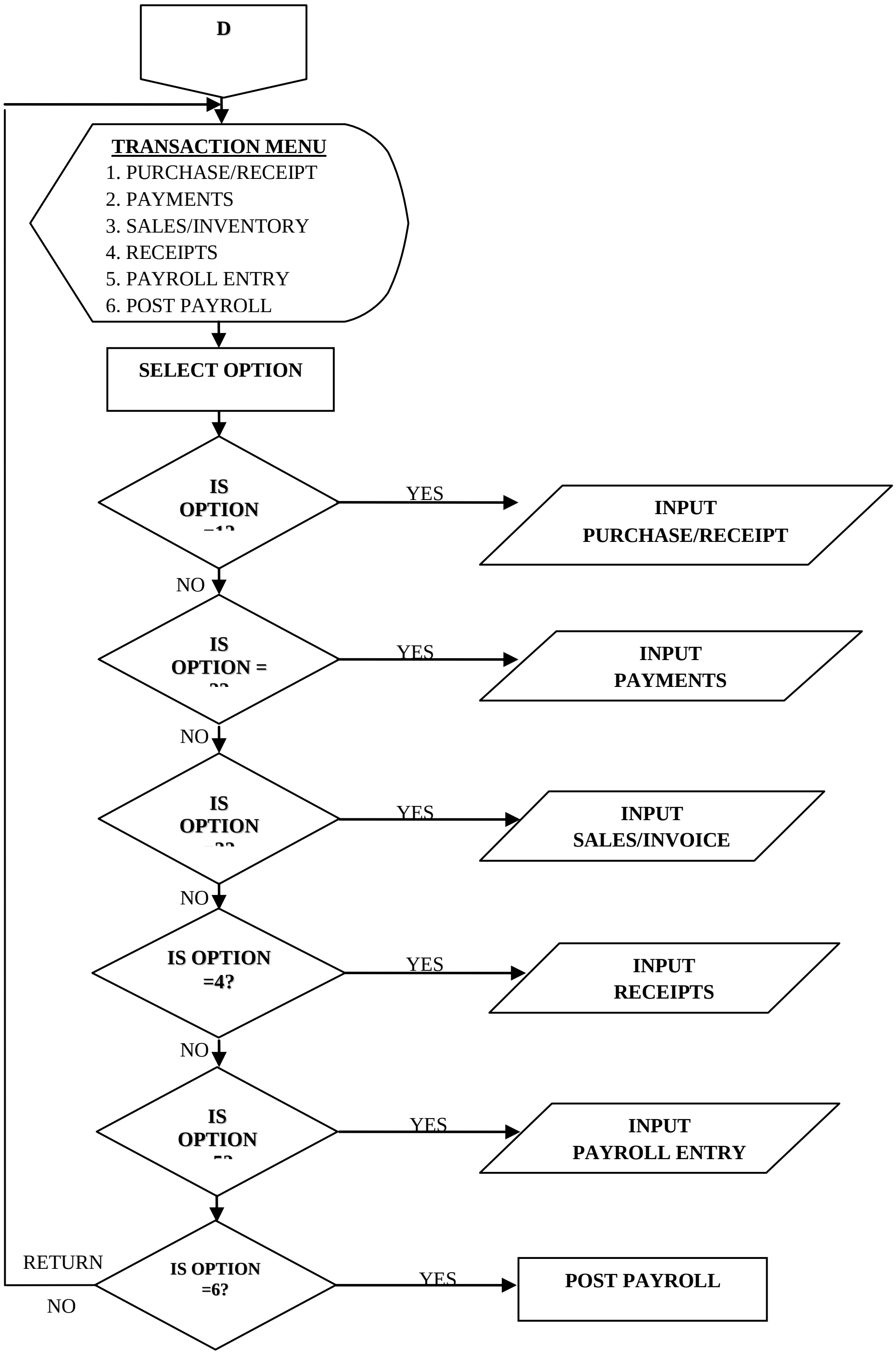


Figure 4.48 Transaction Subsystem Flowchart

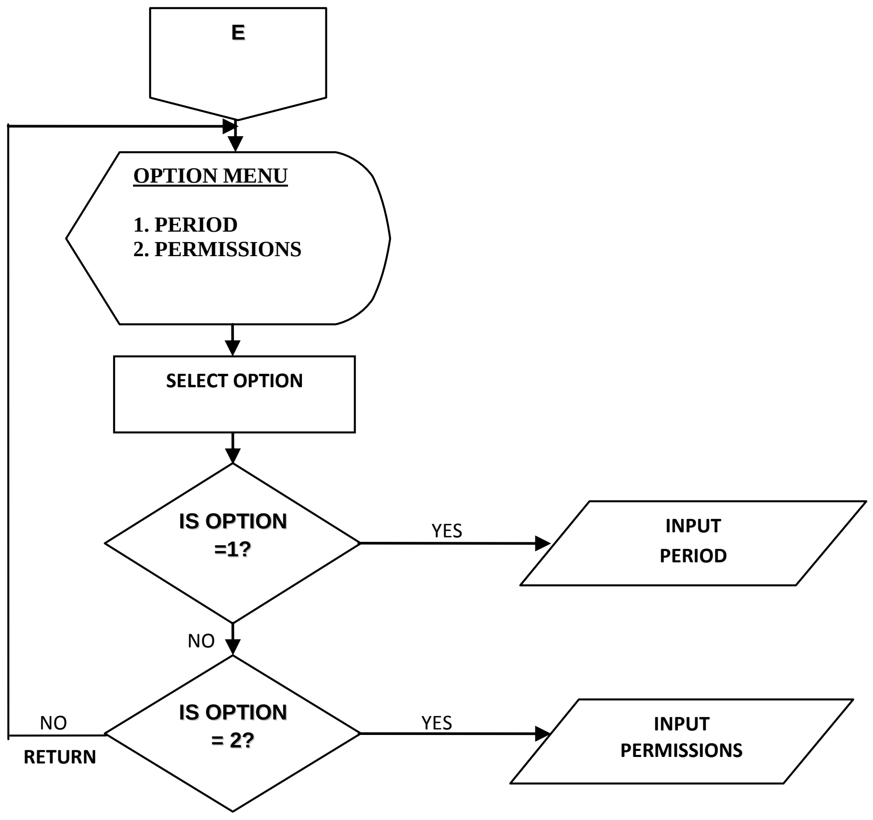


Figure 4.49 Options Subsystem flowchart

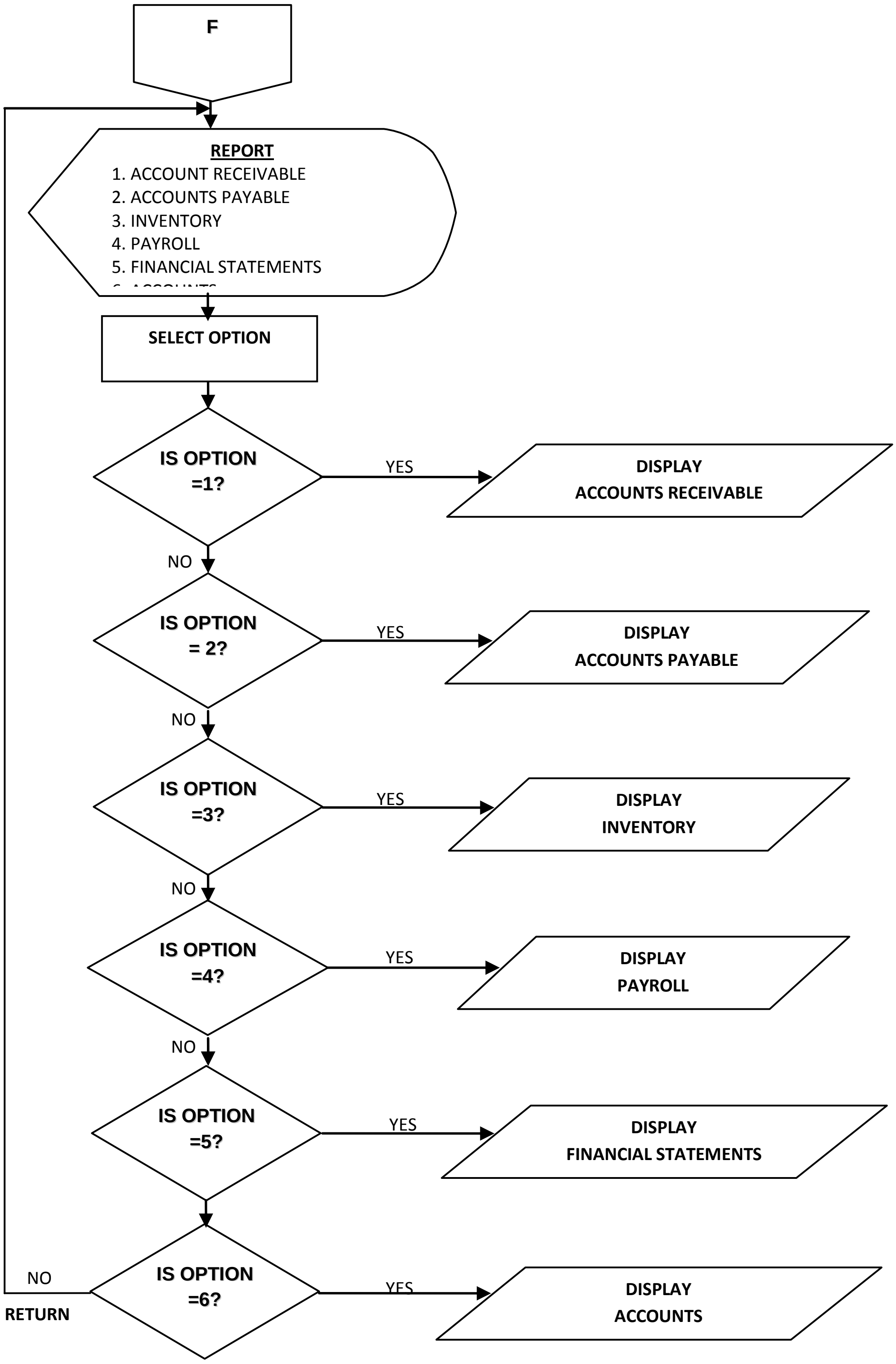


Figure 4.50 Report Subsystem flowchart

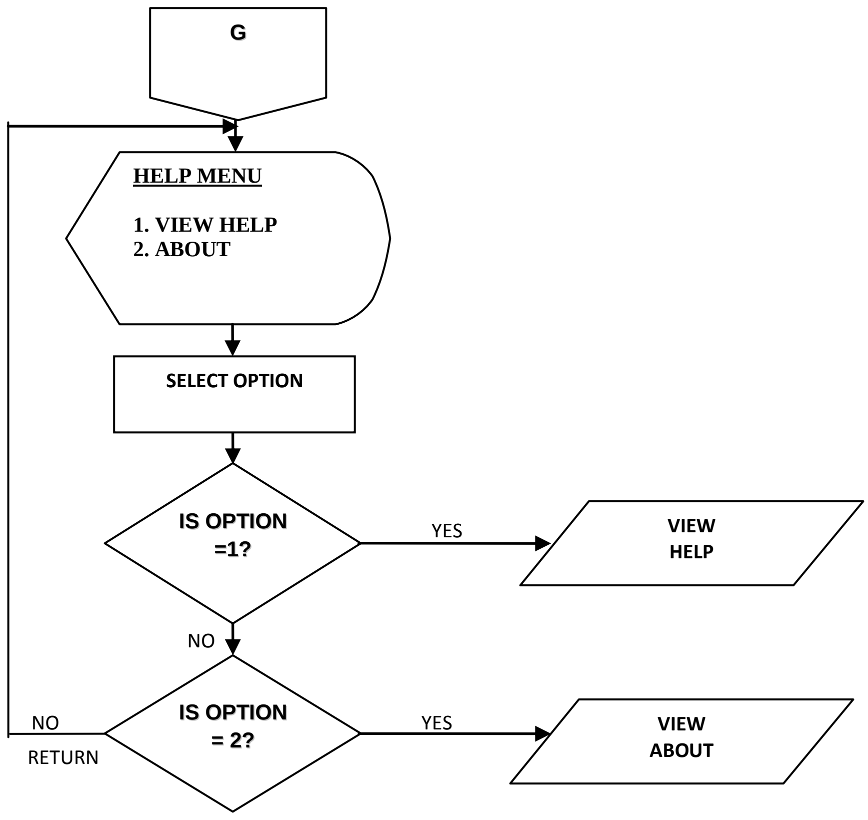


Figure 4.51 Help Subsystem flowchart

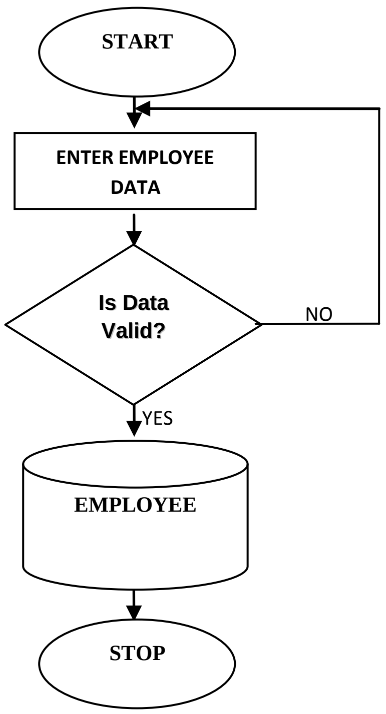


Figure 4.52 Algorithm for Entering Employee Data

4.11. Systems Flow Diagram for the Proposed System

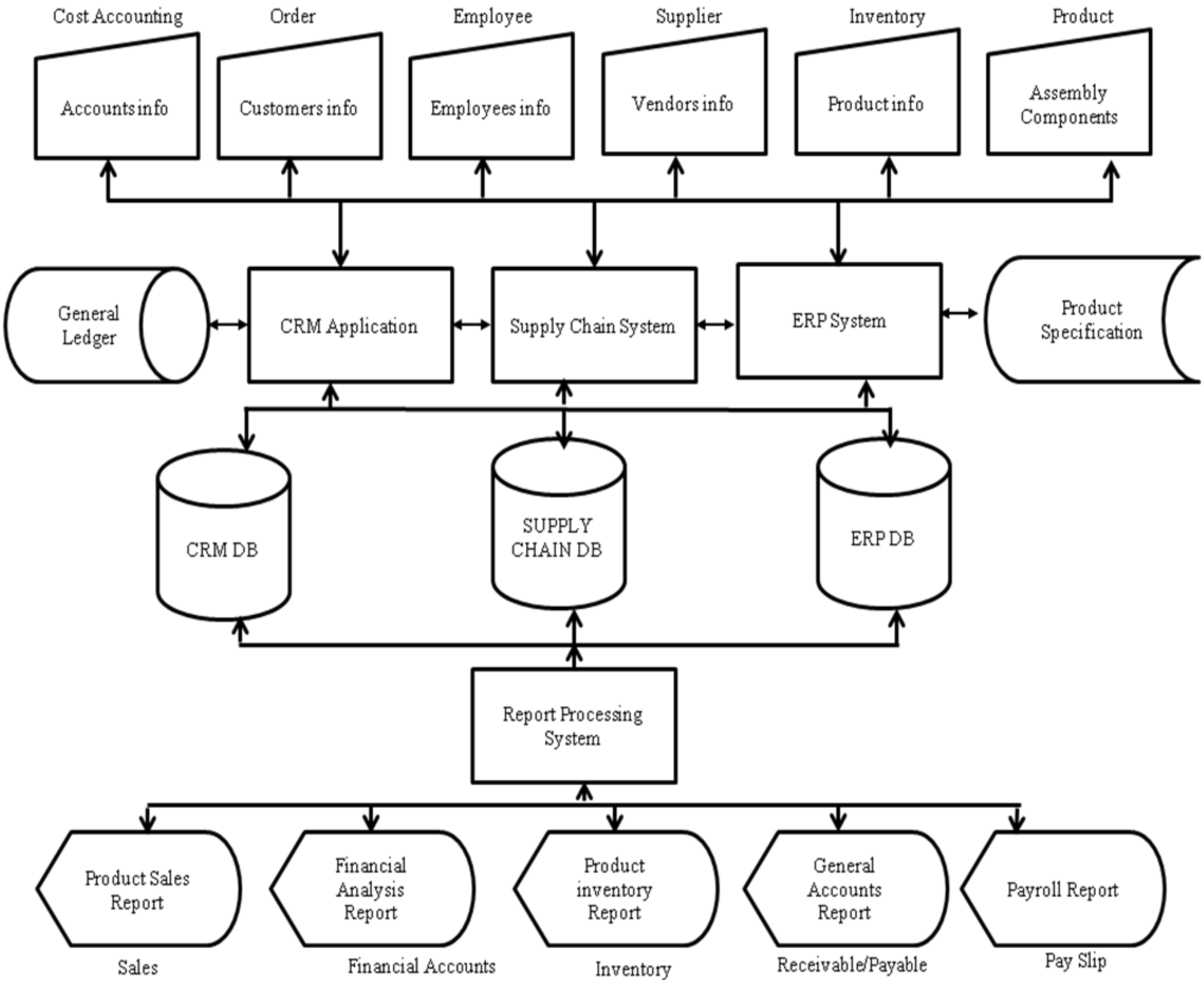


Figure 4.53 System Flowchart for TIPS Application

The Generated Code For This Application Are Available In Appendix I

CHAPTER FIVE

IMPLEMENTATION AND PROGRAM DOCUMENTATION

5.0 Introduction

System Implementation refers to the real live running of the design and developed application. This chapter shows the practical application of the technical reference architecture. This is supported by our methodology. This chapter shows the program modules, what they carry out, how the system can be deployed and the Security, Network, database and operations.

5.1 System Requirements

The following are the hardware and software requirements of the Total Information Processing System (TIPS) bearing in mind that this application aims at reducing cost and helping Small and Medium Enterprises (SME's) streamline their internal operations.

Software Requirements

The minimum software requirements are stated below

- Operating System (Windows XP SP 3, Window 7 (Ultimate, Starter, Home), Windows Server 2003 (32 bit or 64 bit).
- SQL Server 2000 or 2005
- HTTP Servers (Internet Information Services IIS)
- Visual Studio 2005
- Any Desktop Application Servers

Hardware Requirements

- Pentium iv / 1.5GHz Processor
- 514 MB RAM
- Free Hard disk of 2GB HDD
- Standard Mouse for desktop PC's
- Super Video Graphics Adapter
- UPS

- Printer
- Local Area Network (LAN)

5.2 Program development for the SME

5.2.1 Choice and Justification of Programming Language

Visual Basic.Net is one of the languages used in Object Oriented Analysis and Design Methodology. The programming language used for the design and implementation of this software is Visual Basic.Net. Choice for this is based on:

- Criteria for selection
- Characteristics
- Advantages

Criteria for selection

In selecting Visual Basic programming language for development of Total Information Processing System (TIPS), the following criteria were considered:

- Availability: The programming language can easily be made available on any system since it can be easily obtained from software vendors or system users and installed in the system.
- Affordable: the cost of getting the IDE with the compiler is not too expensive. It can be purchased at low cost from vendors.
- System Development Ability: the IDE is interactive and user-friendly thereby making it easy for system users to enjoy working with the package.

Characteristics

- It is an Object Oriented Programming Language
- Data instructions for processing of software are combined in self contained modules called 'Object' entities. These encapsulates properties and operations, message-passing, classes, inheritance and most importantly each of the class are reusable in another program.
- It is event-driven
- It has rich library which enables one to develop programs easily

- It is flexible, allowing the users to customize, like creating your own library.

Advantages

- It helps you develop application that run under windows operating systems
- It is a language for multi-user application
- It aids modular programming approach and makes debugging easier
- It facilitates fast programming development
- Modification can be made in the program classes without affecting the entire program
- Visual Basic is a platform independent language. It is secure, robust and can be compiled.

5.2.2 The System Drivers –Main Menu

The system has a main menu which serves as the system driver. The main menu is as shown in figure 5.2. It contains the operational environment with seven modules as can be seen (Figure 5.2) once you log in into the system. This includes the file menu, the groups menu, the transaction menu, the mail menu, the options menu, the reports menu and the help menu.

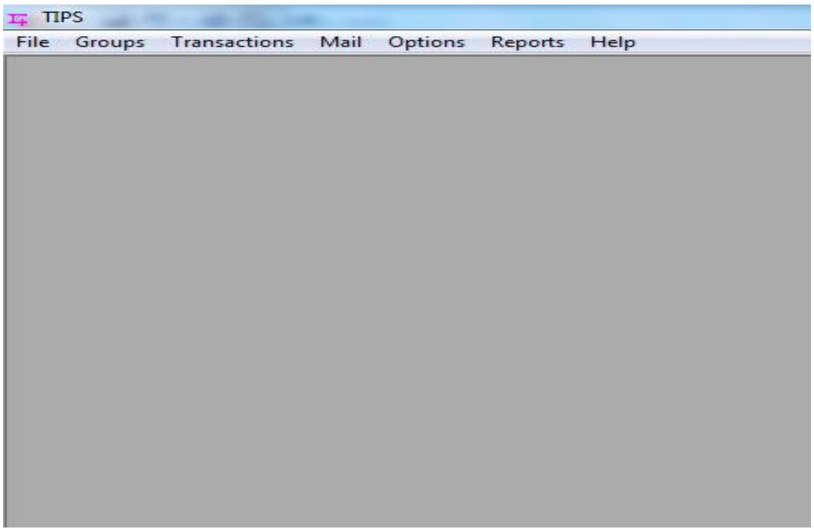


Figure 5.1 TIPS Main Menu

Total Information Processing System

i. File: The file menu allows the administrator to use the sub menus to create user, delete user, login, logoff and close the program

ii. Groups: The group menu allows the administrator to use the sub menus to define the basic data needed to operate the system. These are data that do no change always. The various definitions make up the sub-menu and is defined as the master file. These includes; accounts, customers, employees etc,

iii. Transaction: The transaction menu enables the user to use the sub menus. This is the operational menu that will be used to carry out various operations in the system. It serves as major operational input to the system and is defined as transaction file.

iv. Mail: The mail menu allows the user to use the mail sub menus and this acts as a collaboration menu between the various users of the system. They can send and receive messages over the network.

v. Option: The option menu allows the administrator to use the option sub menus which allow him set transaction periods. Control user access to the functions of the system.

vi. Report: The report menu allows the users to use the report sub menus. This allows management to generate some predefined reports.

vii. Help: This menu provides the interface to the help sub menus. It enable users to access the help about the use of the system.

5.2.3 Implementation of the Sub-Systems

The sub-menus (sub-systems) deal with the collection of data from various sources about the various functions in the system, the storage of such data and the retrieval when needed. These are defined as shown in figure 5.5

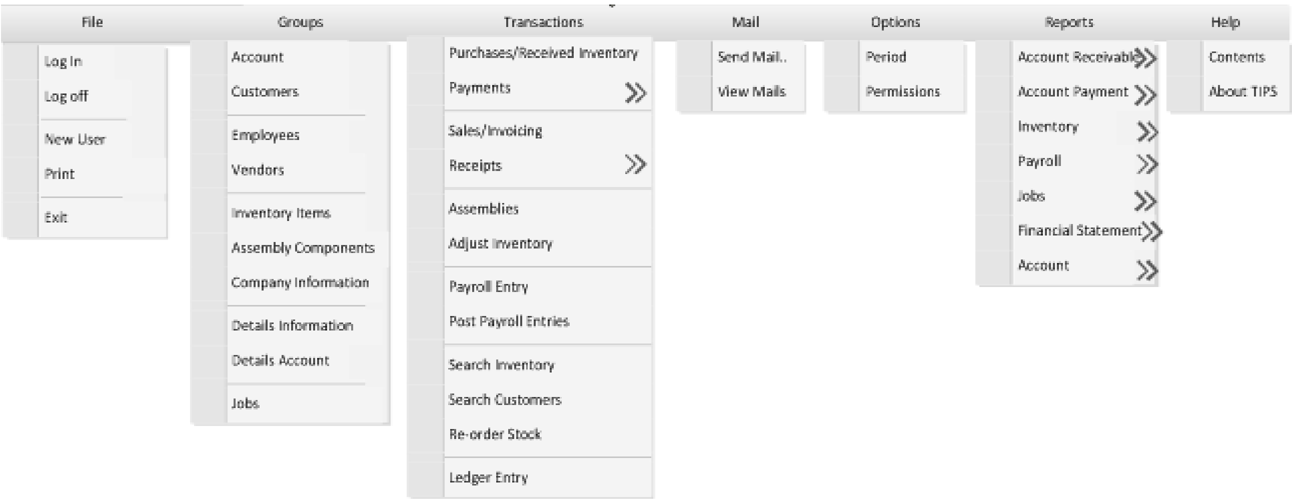


Figure 5.2 TIPS Sub Menu

i. File

- Login: This enables the user to log into the system
- Logoff: This enables you to log out of the system
- New User: This allows the administrator to create new user
- Delete User: This enables the administrator to delete a user

Figure 5.3 File Sub-Menu

ii. Group

- Customers: This enables the input of the initial data about each customer. The data to be entered includes: Customer ID, Name, address, Description and account information.
- Accounts: This enables the setup of the initial account data. The data to be entered includes: Account ID, type, name, description.
- Employee: This enables the setup of the initial employee data. The data to be entered includes: Employee ID, Name, type, address, Payinfo

- Vendor: This enables the setup of the initial supplier data. The data includes: Vendor ID, Company name, Contact Name, Address, Account info.
- Inventory Item: This enables the setup of initial stock data. The data includes; Item ID, Item description, items detail, minimum stock, reorder qty and account information.
- Company Info: This enables the setup of initial company information. This includes; Company name, address, phone no.
- Default info: This enable definition of data that has to do with printing options for customers, vendors,

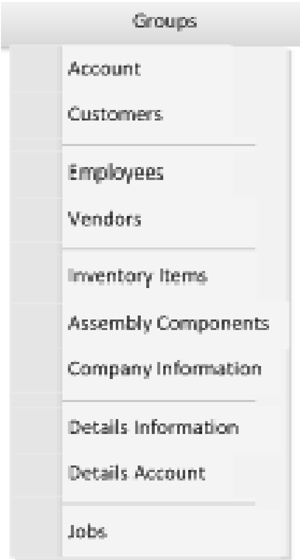


Figure 5.4 Groups Sub-Menu

iii. Transaction

- Purchase/Receive: This enables the entry for purchases done by the company and reception of goods.
- Payments: This enables the company to pay their vendors.
- Sales/Invoicing: This enables the company to make sales and invoice.
- Receipts: This enables the company to issue receipts to customers
- Payroll Entry: This enables the input of employee pay information.

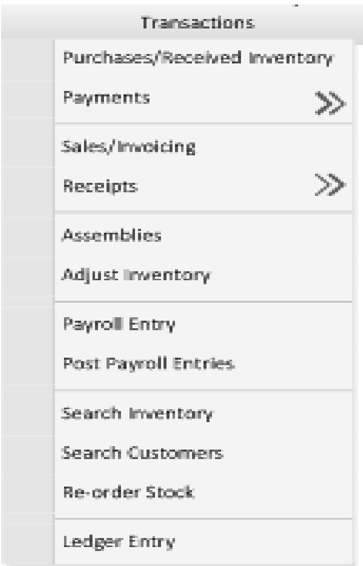


Figure 5.5 Transaction Sub-Menu

iv. Mail

- Send Mail: This allows the users of the system to exchange messages among themselves.
- View Mail: This allows the users to read their messages

Figure 5.6 Mail Sub-Menu

v. Options

- Period: This enables the administrator to create transaction periods, open period and close period.
- Permissions: This enables the administrator to assign functions and rights to a user.

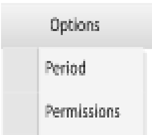


Figure 5.7 Options Sub-Menu

vi. Reports

- Accounts Receivable: This enables the generation of account receivable reports.
- Account payable: This enables the generation of up to date account payable report based on the selected criteria.
- Inventory: This enables the generation of stock report based on selected criteria.
- Payroll: The enables the generation of reports that has to do with employee pay information. This includes: employee list, payroll register etc.
- Financial Statement:: This helps to generate financial statement reports. This includes; Trial balance, income statement, balance sheet.

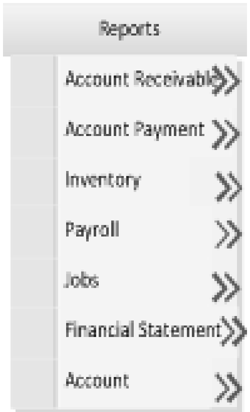


Figure 5.8 Reports Sub-Menu

From the point of initial creation, customer can place order and purchase products. Customers should have discount and credit limits attached to their account. The system can be used for monitoring customers purchase and credit limits. This information will be useful to determine customer’s eligibility to take purchase more products.

The transaction module enables the user to make sales, receive items and can also be use for employee salary.

A number of reports can be generated from the system from ready-made reports and ad-hoc reports. The system also includes an enquiry screen to list employees, customers, and vendors. The Option module includes facility for setting up users and entering system parameters

The detail interface is as shown in Appendix III.

5.2.4. Database Implementation – Microsoft SQL Server 2005

SQL Server is a software package that enables the creation, maintenance and management of database. SQL Server is a Structured Query Language (SQL) based, client/server relational database. Each of these terms describes a fundamental part of the architecture of SQL Server.

Database: A database is a storage place for data. The user runs an application that accesses data from the database and presents it to the user in an understandable format.

Relational Database: There are different ways to organize data in a database but relational databases are one of the most effective. Relational database systems are an application of mathematical set theory to the problem of effectively organizing data. In a relational database, data is collected into tables (called relations in relational theory).

Structured Query Language (SQL): There are several different languages that can be used to manipulate relational databases. The most common of the languages is SQL. The American National Standards Institute (ANSI) and the International Standards Organization (ISO) have defined standards for SQL. Data within a database can be retrieved via SQL that is based on Relational Algebra.

Client/Server: In a client/server system, the server is a relatively large computer in a central location that manages a resource used by many people. When individuals need to use the resource, they connect over the network from their computers, clients, to the server.

SQL Server's specific design goals were speed, robustness and ease of use. To improve the performance, SQL Server was made as a multithreaded database engine. A multithreaded application performs many tasks at the same time as if multiple instances of that application were running simultaneously. Multithreaded applications have a lower overhead cost, when compared with multi processed databases. In being multithreaded, SQL Server has many advantages. A separate thread handles each incoming connection with an extra thread that is always running to manage the connections. Multiple clients can perform read operations simultaneously, but while writing, only the clients that need access to the data being updated are held. Even though the threads share the same process space, they execute individually. Because of this separation, multiprocessor machines can spread the thread across many CPUs as long as the host operating system supports multiple CPUs. Multithreading is the key feature to support SQL Server's performance design goals and this is the core feature around which SQL Server is built. SQL Server has other features but the most attracting features are cost and performance.

The SQL Server Environment

This contains the database instance and the tables that store the data from the application.

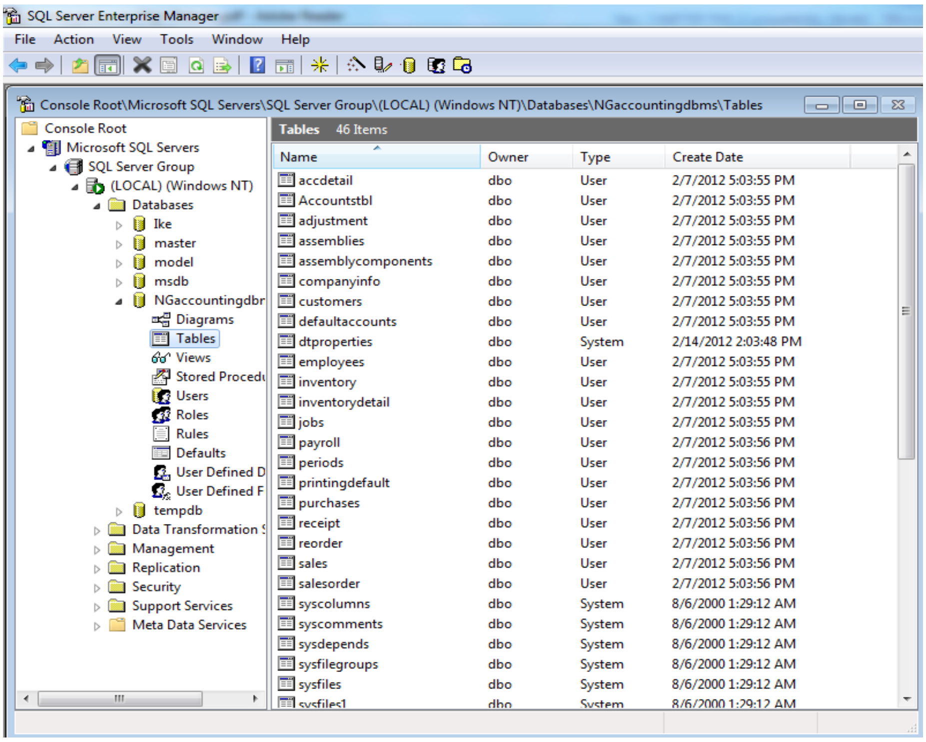


Figure 5.9. SQL Server Enterprise Manager

5.3 Program Testing

5.3.1 Software Sub-System Testing

The software subsystem testing is divided into seven modules:

The Log In module, Groups module, Transaction module, Mail module, options module and Reports Module.

1. The Log In module was first developed and tested to ensure that only authorized users have access to the system. This tested ok.

2. Groups menu module was also developed and tested to ensure that the connection to each data access is displaying the appropriate access addressed. And to be sure that the data is stored and can be retrieved whenever needed.

3 Transaction Module was also tested to ensure that the inputs to each transaction file are stored in the database and the record of any customer whose code is searched is displayed. It was also tested to ensure valid data are stored for each transaction and that appropriate inputs are made whenever there is a new transaction.

4 Mail Module was also tested to ensure that users can send and receive mails when there are working with the system.

5. Option Module was tested to ensure that when a period is set, it is applicable to all modules working with account period and also to make sure that access given to any user is the access enabled for the user.

6. Report Module was tested by making sure that the needed data is generated whenever a report parameter is submitted at different time interval. It was also tested to be sure that only valid data is generated by the reports.

The System testing is tabulated as shown in table 5.1

Table 5.1 System Testing

S/No	User Interface	Expected Result	Actual result
1	Group Menu – Customer	It should be able to store data, retrieve data and delete data	New inputs were made on the system. It was able to add, retrieve and delete records. Tested ok.
2	Transaction – Sales	It should be able to store customer information, update stock, check credit limit and post transaction	Transaction data was saved and posted. Customer status was checked and stock data updated.
3	Transaction – Purchase	It should receive and save inventory data, update stock and post transaction	The stock data was stored, stock updated. Accounts data updated

4	Mail – Send Mail	It should be able to send mail to a selected employee	Mail was sent and reply received from employee
5	Options – Permissions	It should be able to grant a user access to required functions of the system.	A user cannot perform a function that was not assigned to him or her. Tested ok.
6	Reports – Inventory	It should be able to generate inventory data showing the status of all the items	Report was generated showing each item with the stock quantity.
7	Reports – Payroll	It should generate data showing employee information with status.	Report was generated showing a list of employees.

5.4 System Conversion

5.4.1 Change over Procedures

There are various changes over procedures in system implementation. The four most widely used are:

- Direct change over procedure
- Pilot change over procedure
- Parallel change over
- Phase change over

i. Direct changer over

Direct change over is a system implementation where the company automatically changes from manual procedure to the new system. This is not always considered as the best option because the new system may not be meeting up to expectations and may need some modification

ii. Parallel change over

This involves the use of the new system and the old one concurrently to know which is best, in order to avoid inconsistency in managing the system for a give period of time, for optimal performance

iii. Pilot change over

Pilot change over involves the use of the new system and old system. Meanwhile, the new system is implemented in parts for some period of time to check the performance before all the system will be implemented.

iv. Phase change over

This change over procedure adopts phase implementation of the system. The conversion to the new system is carried out in phase.

5.4.2. Recommended Change Over

Having considered all the change over procedures, we recommend the use of parallel change over in order to compare the effectiveness of the new system in line with the normal operational mode in the existing company; to give room for modifications should the need arises.

5.5 Security and Access Control

Access to the system is menu-driven and dependent on the access level granted to each user. There are three levels of users within the system. They are:

a. The Supervisor/System administrator level, which is valid for all modules within the system

b. The Data Entry level, which is valid for all modules relating to data entry in the database

c. The Executive user level, which is valid for viewing reports To gain access to the appropriate module, a user enters his/her user name and password given by the System Administrator from the main entry screen. The users are required to change their password at first logon; they can also change their password at any other times.

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Specific requirements in this area could include the need to: Utilize certain cryptographic techniques

- Keep specific log or history data sets
- Assign certain functions to different modules
- Restrict communications between some areas of the program
- Check data integrity for critical variables
- Later version of the software will incorporate encryption techniques in the user/license authentication process.

The software will include an error tracking log that will help the user understand what error occurred when the application crashed along with suggestions on how to prevent the error from occurring again. Communication needs to be restricted when the application is validating the user or license.

5.6 Training

5.6.1 Software Training Support

Well packaged software training will be provided by the organisation in order to equip the staff who is to be in-charge of the system on how to manipulate business activities with the application. This is necessary so as to avoid data loss while working the new system and also avoid invalid entry. However, documented manuals are made available for better understanding of the entire system processes and procedures.

5.6.2 Staff Training

Before a staff of an individual uses this application, the person has to be trained on how to use this application. The system/database administrator should be trained first. He will in turn train some other staff of the organisation that will be operating the application.

5.7 System Documentation

System documentation is the task of documenting the internal design of the software, the systems requirements and constraints and the operational procedures of the system for the purpose of further maintenance and enhancement. It could be liken to the product manual that produces details of the processes, operations, problem solution mechanism. This is usually printed out and distributed to system users containing the design of the new system, to enable users or operators to know the correct ways of using the system.

5.7.1 Installation of the Software

Total Information Processing System (TIPS) was built using Visual Basic.Net and SQL Server Database. The steps below show how to configure the TIPS.

Steps

The software was stored on a CD with an executable file

- Insert the CD in the CD-ROM drive
- Double click on the application icon as compact disk displays its content
- The application installation is menu driven and interactive, follow the step by step process until you get to the end of the application installation.
- At the SQL server, create a database with name ‘NGAaccounts’ and assign a byte size of 50MB for the Initial block.
- No need to create any table. Once you run the ‘TIPS’ applcation for the first time, it will create all the tables, queries and views at the database.
- Configure the Registry to create a handle for ‘TIPS’
- Run the application.
- The splash screen will appear followed by the main menu, Click on Login menu and enter a correct username and password.
- The main menu displays where other activities can be executed
- The menus includes: File, Group, Transaction, Reports, Options and Help menu
- Click on any of the menus to see the sub menus which will enable you to execute any operation you desire to perform.

5.7.2 Operational Procedure

To operate this application, some steps need to be followed. These steps are:

- Open the application by double clicking the TIPS icon on the Desktop.
- Enter a valid username and password
- Navigate through the menus.

To create new users of the system

- From the main menu, click on the file sub menu; select New User from the drop down menu.
- Enter the new user data and click Ok
- From the Option Sub menu, select Users and Roles, Select the New User created and assign roles depending on the task he/she will be performing.
- You can also Delete any user from this sub menu

To Setup data (Update Static Records)

- From the main menu, click on the Group sub menu, select the menu form containing the information you wish to update.
- Enter the information required using the provided input controls, click Ok to save and the close the form

To Make Transaction (Daily activities)

- From the main menu, select the Transaction sub menu, select the menu form containing the information you wish to update.
- Enter the information needed using the provided input controls, click ok to save and then close the input form.

To Generate Reports

- From the main menu, select Reports sub menu, select the menu containing the report you wish to View or Print.
- Supply the necessary Parameter as may be required in the Criteria form provided.
- Click Ok to generate the report.
- You can just View the report as displayed on the screen or click on Print icon to Print the report.

5.7.3 User's Guide

Users of the software should take the following as a guide:

Log In

The system starts with login form where the registered user can enter user name and password to be able to access the system. Figure 5.5 shows login form.

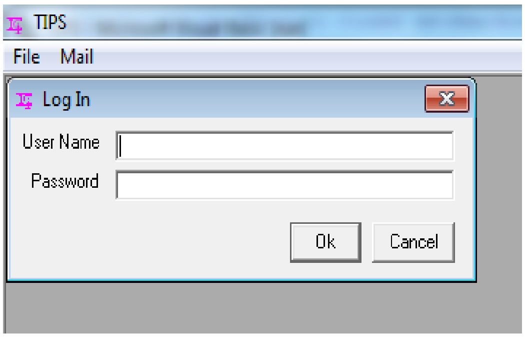


Figure 5.10 Log in

Once the user log in successfully, the main menu interface will be displayed. This contains the modules that user can work with. The main menu is shown in figure 5.12.

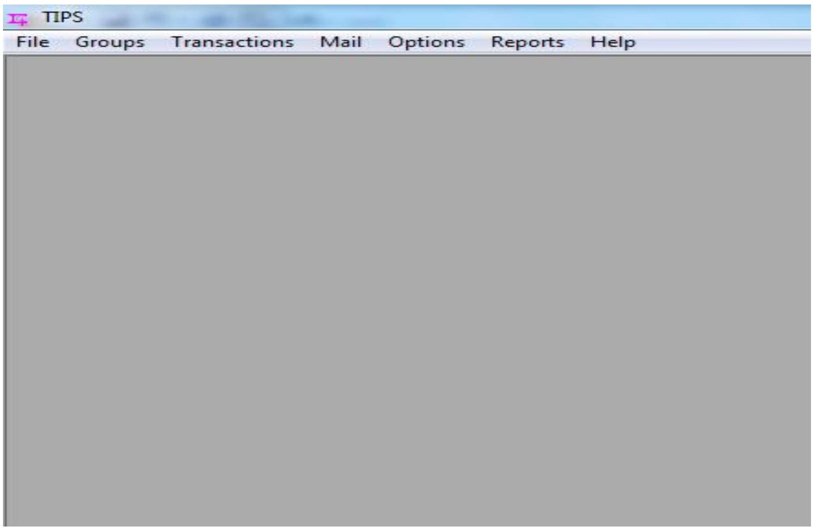


Figure 5.11 Main Menu Interface

The main menu contains seven (7) sub-menus. These are file menu, groups menu, transaction menu, mail menu, options menu, report menu and help menu.

File Menu

The file main sub program allows the administrator to create new users and delete a user. It allows the user to log in and log out.

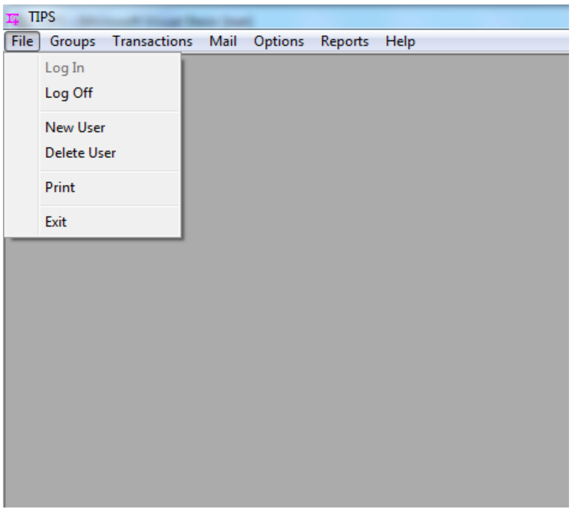


Figure 5.12 File Menu Interface

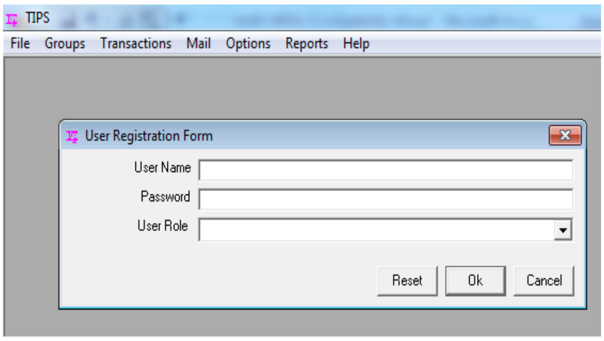


Figure 5.13 User Registration

Groups Sub Menu

The groups sub menu allows the user to create initial startup data for the system. These data includes;

Initial account data, customer data, employees, vendors, inventory, company information and some default information about the customers and vendors.

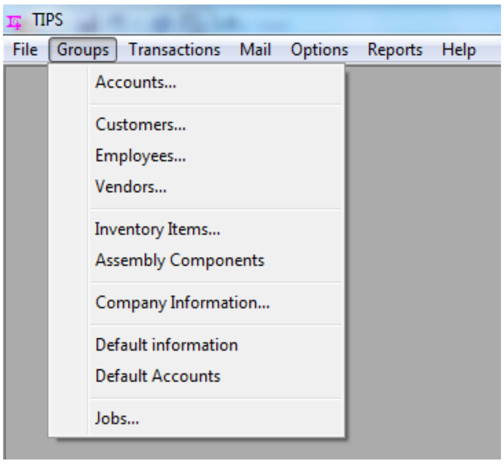


Figure 5.14 Groups sub-menu interface

Accounts form

Figure 5.15 shows Accounts form which contains details of Accounts information during initial setup.

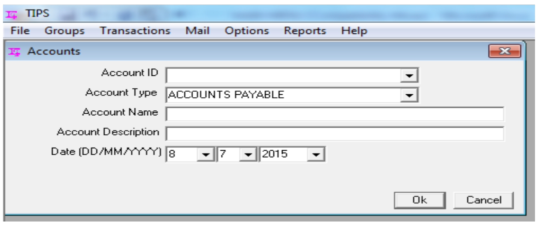


Figure 5.15 Accounts form

Customer Form

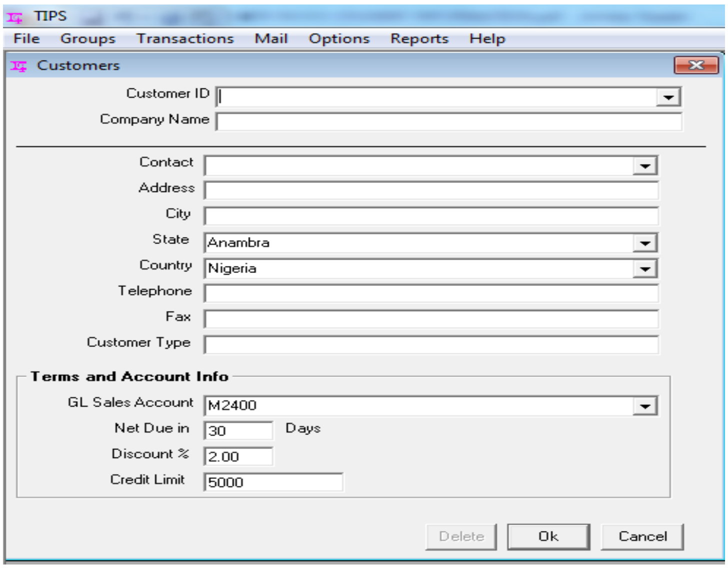


Figure 5.16 Customer form

The customer form contains detail information about a customer. This includes; Customer id, Company name, address, terms and account information.

Employee form

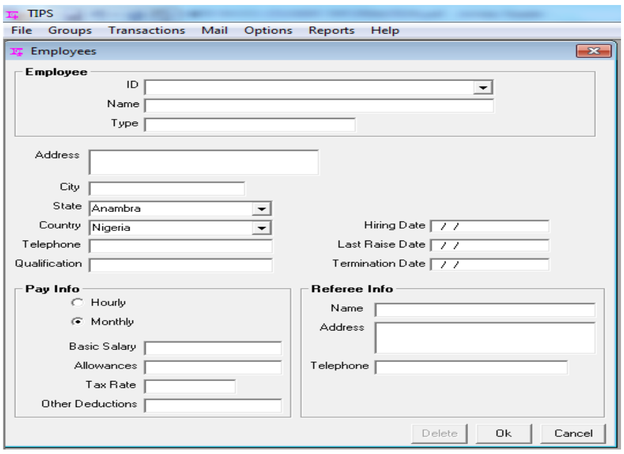


Figure 5.17 Employee form

The employee form contains detail information of an employee. If there are changes in the data, it will be updated

Vendor Form

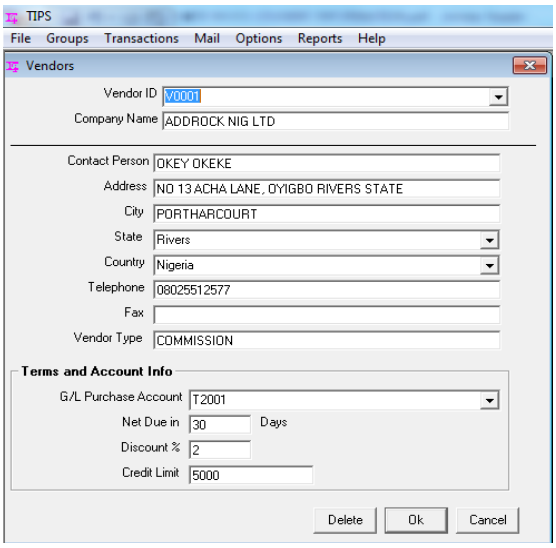


Figure 5.18 Vendor form

This form consists of Vendor id, company name, contact person and address etc. If any changes will with respect to change it will be updated.

Inventory form

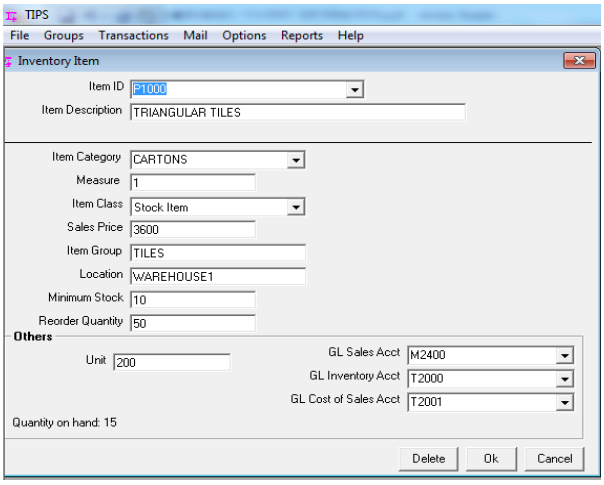


Figure 5.19 Inventory form

This form consists of item id, item description, category etc. any update will be done here and the data will be updated.

Transaction Sub-Menu

This sub-menu displays the transactions to be performed by the user. This is the daily operations that will be performed with the system. This includes product reception, sales, accounts update, assembly components, payroll entry for employees, issue receipts etc. this is basically user module.

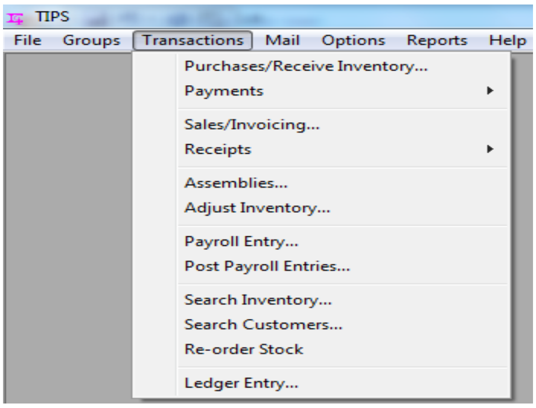


Figure 5.20 Transaction sub-menu

Purchase/Receive Inventory Items

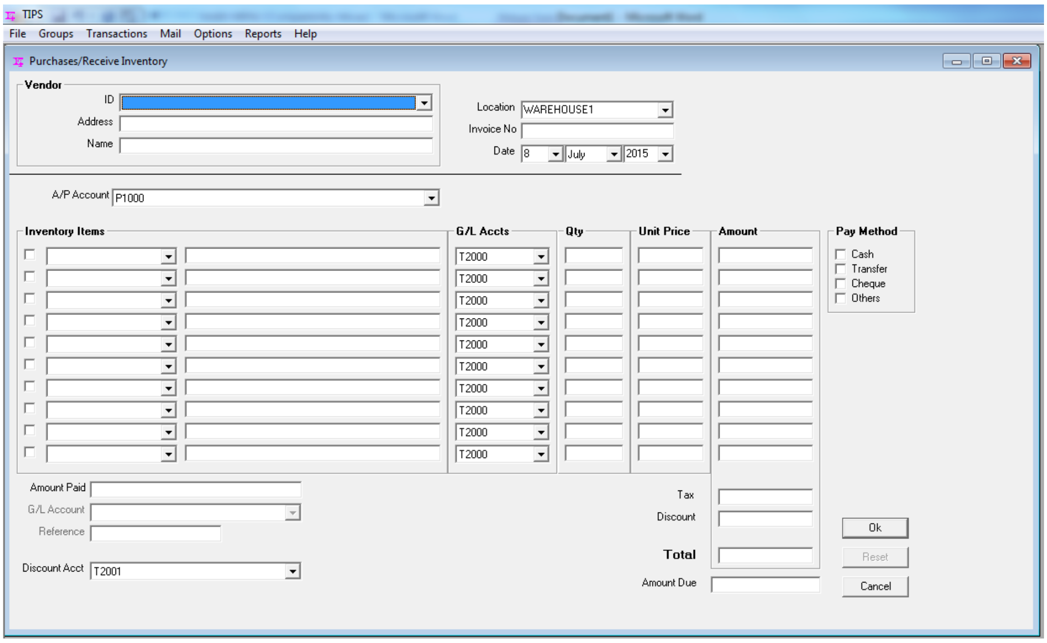


Figure 5.21 Purchase/Receive Inventory

This allows the user to receive inventory from the vendor or from production. When updates are made to the system, the inventory will be update as well as the necessary financial books (accounts payable). The form contains such information as vendor id, item location, Accounts information etc.

Sales/Invoicing form

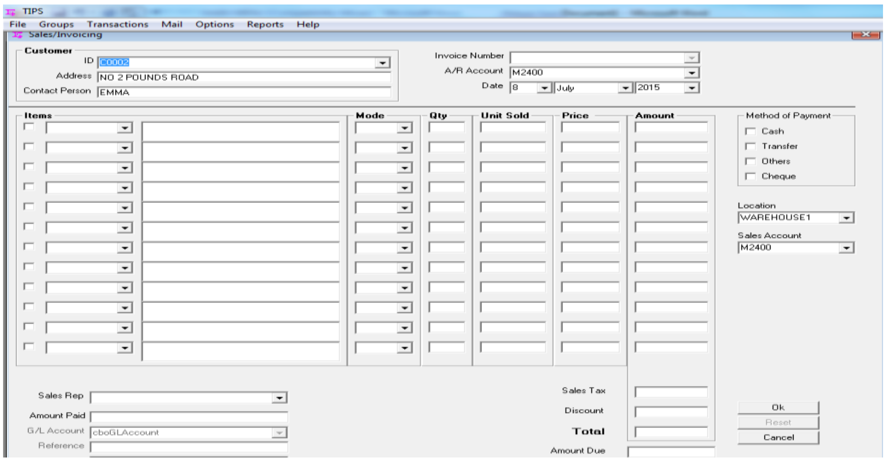


Figure 5.22 Sales form

This allows the user to sell products to customer. The information contained in this form are; customer, invoice, item accounts to be updated. Once sales are made, the inventory will be update to reflect the current status.

Payroll Entry

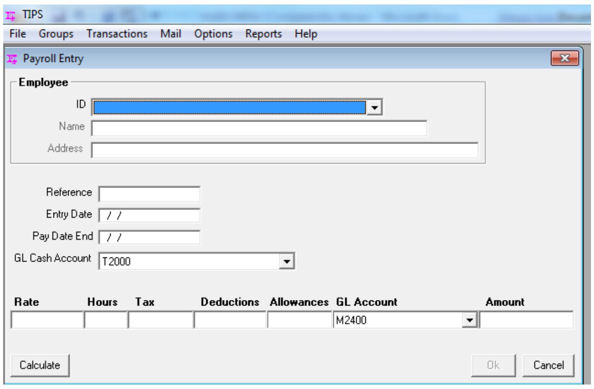


Figure 5.23 Payroll Entry form

This form enables the user to update employee payroll data. The information on the form includes; employee id, status, accounts information etc. when update are made to employee payroll data, the data will be update for payroll processing.

Build Assembly form

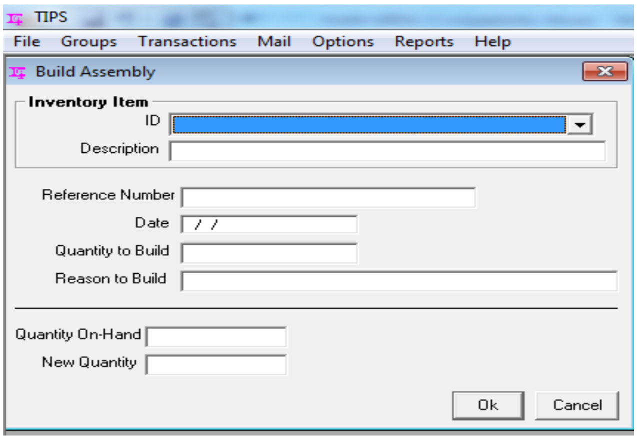


Figure 5.24 Assembly form

This form allows the user to build assembly – combine two or more components to produce a product. The information in this form are inventory items, reference, quantity etc.

Report Sub-Menu

The report sub-menu allows the user to generate one or more reports. This includes accounts receivables, accounts payable, inventory listing, and payroll.

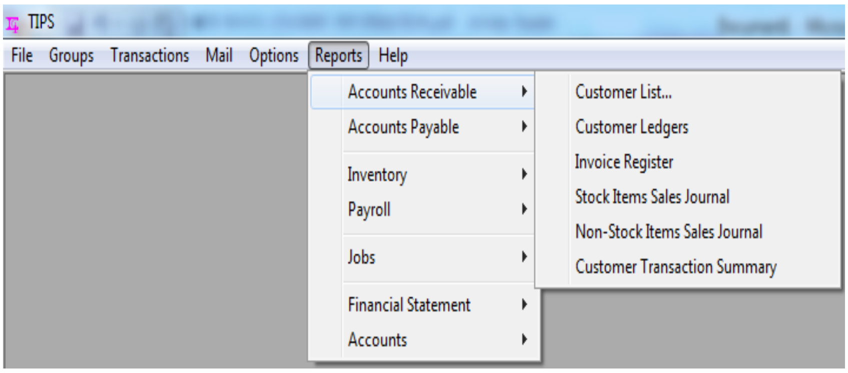


Figure 5.25 Accounts Receivable

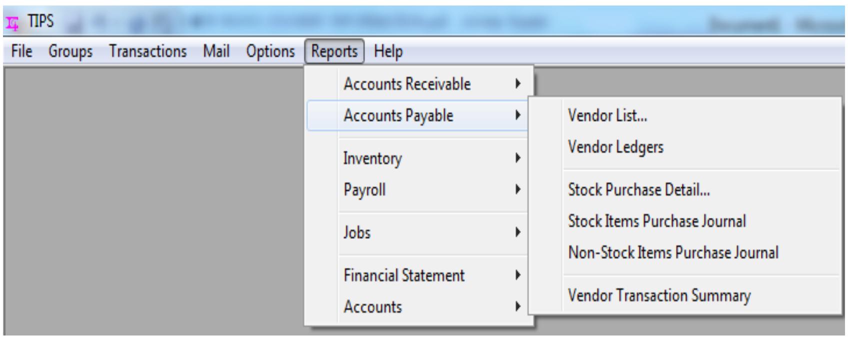


Figure 5.26 Accounts Payable Reports

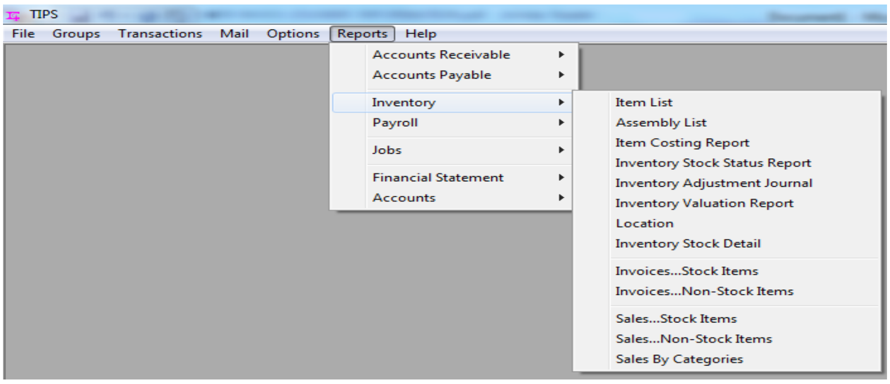


Figure 5.27 Inventory Reports

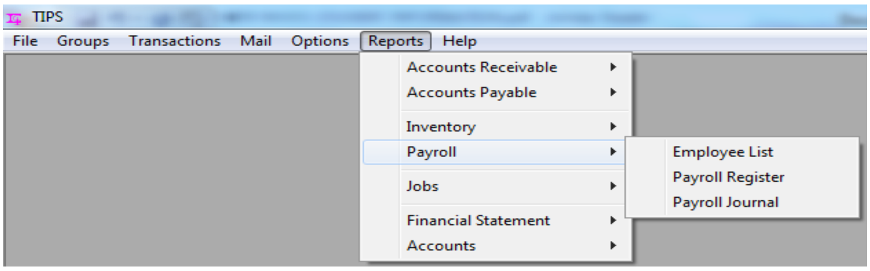


Figure 5.28 Payroll Reports

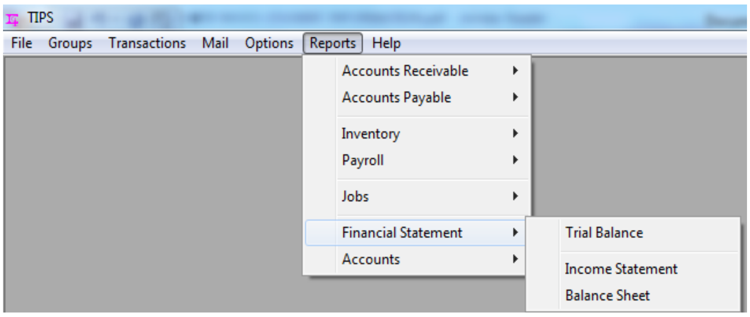


Figure 5.29 Financial Statements

5.8 Continuous Improvement/Maintenance

This implies continuously improving the system to achieve new enterprise objectives or new business process requirements. It deals with the necessary changes in a system. Enhancement is needed for the system to be able to meet the medium and long term improvement projects and to define new processes or changes in the processes.

The following highlight the need to periodically maintain the total information processing system.

- To make sure that all the systems needed to configure are done and running smoothly.
- To efficiently trace user logins which is not free for all access

- To enhance the security mechanism in order to protect hackers from gaining accesss to enterprise information
- To check the efficiency and efficacy of the help menu
- To make sure the application is generating a flexible report on the general system aspect

Maintenance should be a continuous activity in order to ensure that the system is working properly. Updates of the system could be undertaken based on the current state of the art technology.

5.9 Architectural Control – Implementing Governance and Security View for the SME

The Governance and security view defines and enforce architectural control. The governance and security involves policies, standards, and metrics for the implementation of the enterprise architecture. Objectives, principles and capabilities that govern this view are presented in table 5.2.

Table 5.2 Objectives, principles and capabilities of Governance and security View

Governance and Security View	**Name**	**Description**
Objective	Enforce Architectural Control and define policies for implementation	Define the best standard for software and hardware platform
Principle	Choose key technologies and standardize	Minimum set of standards, policies and monitoring
Capabilities	Standards and policies Standardize software platform Standardize hardware platform Single sign-on based on the security	Domain in standards and policy making. Process monitoring. Domain in security management

The objective of the governance and security view is “Enforce architectural standards and control”, the capabilities for the management of governance and security are “Set standards for hardware and software platform, make policies and provide a single authentication for applications. Some of the standards are present in table 5.4.

Table 5.3 Basis for Governance and Security View

	Name	**Description**	**Custodian**	**Domain of Standard**	**Support for business processes**
Standards	Operating system	System software to be used	IT executives, IT and Business Architects and Analysts	Standardize the type of operating system to be used	Governs the development of IS
	Development language and database	Application programming languages and database management system	IT, IT executive, Architects and analysts	Standardize the development platform	Governs the development of IS
	Hardware Platform	Systems and server	IT	Standardize the hardware platform	Governs the development of IS

CHAPTER SIX
TESTING AND EVALUATION

6.0 Introduction

Testing and evaluation (T & E) is the process by which a system or components are compared against requirements and specifications through testing. The results are evaluated to asses' progress of design, performance, supportability etc. developmental test and evaluation (DT & E) is an engineering tool used to reduce risk through the development cycle. Operational test and evaluation OT & E is the actual or simulated employment by typical users of a system under realistic operational conditions.

Testing is a mechanism to assure quality of a product, system or capability. To be effective, testing cannot occur only at the end of a development. It must be addressed continuously throughout entire life cycle.

Test and evaluation involves evaluating a product from the component level, to stand-alone system, integrated system and if appropriate system-of-system and enterprise.

6.1. Objectives of the system Test

At a high level, the system test intends to prove that;

i. The functionality delivered by the system, is as specified by the business in the business design specifications document and the requirement documentation
ii. The software is of high quality; the software will replace/support the intended business functions and achieve the standards required by the enterprise for the development of new system
iii. The software delivered interfaces correctly with existing system including Windows 7.

6.2. Overview of the New System (TIPS)

The Total Information Processing System will enable;

i. Integration of the major business functions of the enterprise.
ii. Synchronization of the business processes
iii. Removal of any legacy system

iv. Introduction of new technologies
v. No constraint on location of caption
vi. Capture of transaction for other processing system

The new system will do the following:

- Provide the users with menus, directions & error messages to direct him/her on the various options.
- Handle the update/addition of employee information, Stock, purchases, sales, product assemble.
- Print various reports which includes invoices, sales summary, purchase summary, accounts information
- Create a transaction file and transfer the file to the excel worksheet for further analysis.
- Run on Microsoft Network using IBM compatible PCs as data entry terminals

6.3. Testing Process

The best practice of testing was adopted which to test each subsystem separately. The lowest units were tested and retested after combining with related units.

The levels tested are ;

i. Unit – the procedures, functions, methods, individual input interfaces were tested at this level
ii. Module – the packages, classes, individual modules were tested
iii. Sub-system – collection of related modules, cluster of classes, connection pathe were tested at this level.
iv. Acceptance – the whole system was tested with both real and test data. Both Alpha and Beta testing were employed.

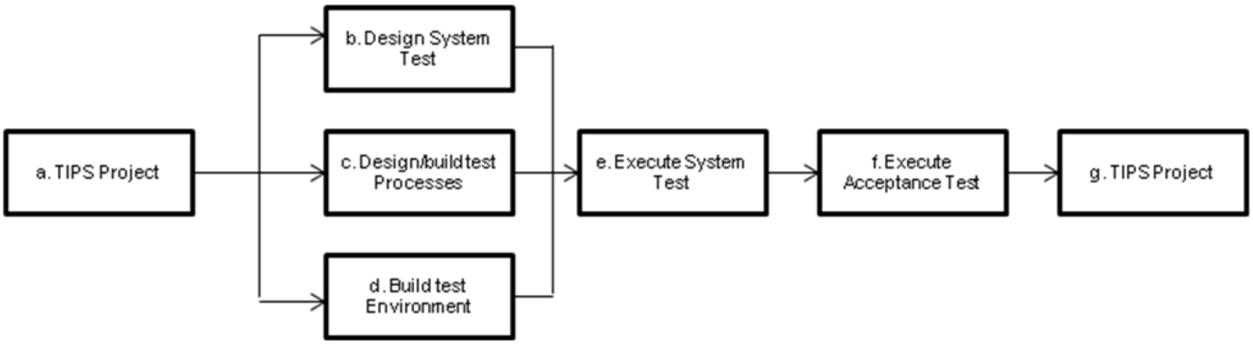


Figure 6.1 Test Process Approach Adopted

6.4. Test Data

The test data was made from order placed by two different customers for the supply of goods. The respective customers' were already known customers of the enterprise. The criteria for the ordering are;

i. The customer must be a registered customer with known business name and business address.
ii. The order amount must be less than the customer risk level: order amount must be less than customer credit limit.
iii. Stock quantity must be higher than the order quantity else there will be added production.
iv. A new customer must be registered before any transaction.

Note: The Order Process was used as the test data because the process cuts across all the business units including sales, production, accounts and shipping.

Order Processing

Two orders were received from the two different customers as follows;

Order 1: Konas Nig. Ltd

Customer ID: C002

Address: No 2 Pounds Road, Aba

S/N	ITEM CODE	DESCRIPTION	QUANTITY
1	P1000	Triangular Tiles	20
2	P1010	Block Tiles	10

Order 2: ADDRock Nig Ltd

Customer ID:C0001

Address: No 2 Edidi Lane, Idumota, Lagos

S/N	ITEM CODE	DESCRIPTION	QUANTITY
1	P2000	Circular Tiles	10

2	P3000	Irregular Tiles	10
3	P1000	Triangular Tiles	15

The orders were received, entered into the system and processed as follows;

1. Order Input

   Order 1

2. Invoice Generation

At each stage, a critical analysis as performed to make sure that the output data of the system is correct, consistent, of high data integrity and met the client's requirement.

It was found that;

i. The system meets the clients requirement
ii. The processing is flexible and efficient
iii. The output is consistent and correct
iv. It is easy to use
v. It generates all the required reports (business document).

6.5 Evaluation of AIMES Model

The AIMES model was evaluated through a research (case study) process. The evaluation was done to determine whether the model supports the essential EA dimensions (section **4.3.1**) and meets the requirements for EA for SMEs (section **2.8.1.1**).

6.5.1 AIMES Support of Essential EA Dimensions

The AIMES model conforms to the essential dimensions of EA frameworks as identified after the review of important EA frameworks in chapter 2 and defined as AIMES Abstraction Levels in (section 4.3.1).

• The AIMES model covers and integrates the four essential EA focuses: *why* through the Requirement/goal viewpoint (Contextual abstraction), *who* through the actor/device viewpoint (Physical abstraction), *how* through the business process viewpoint (logical abstraction), and *what* through the data/capabilities viewpoint (Conceptual abstraction). Relationships are defined to relate concepts from different viewpoints.

• The AIMES model *blends* the four EA layers (*business*, *IS*, Data, *IT*) by providing *Actors* for each layer (*Human Actor* / *Role*, *Data Actor*, *Software Actor*, *Device*) and enabling the other four viewpoints to be related to it. *Requirements*, *Business process*, and *data objects* could also originate from the four different EA layers, as seen in the EA model of the SME during the research.

• The AIMES model provides a means to analyze the strategy space without worrying about any constraints of processes beforehand, as it separates *Goals* from processes.

6.5.2 Meeting the Requirements for EA for SMEs

The AIMES model conforms to the EA requirements for SMEs listed in section **2.8.1.1**:

1. By providing a means to analyze the SME by using a model, control was increased for the CEO.
2. By conforming to the essential parts of EA frameworks, a holistic overview was provided.
3. Since the AIMES model is based on strategy and goal refinements, the requirements regarding objectives are fulfilled.
4. This requirement (Suitability for the target audience) was split up into SME-specific requirements (see further).
5. Since AIMES is based on the essential dimensions from EA frameworks used for modelling enterprises, it provides an enterprise overview.
6. Since AIMES is made up of different EA layers used for modeling enterprises, a high-level overview of an entire enterprise can be shown in a single integrated and well-defined model.

Further: As the fourth requirement of EA is related to SMEs as a target audience (and, more specifically, to the CEOs or managers of SMEs), the requirements for adoption and successful use of IT in SMEs can be evaluated (section **2.8.1.2**):

4.1. To allow the CEO to work more efficiently, the AIMES model is kept to the bare minimum (e.g., a comparison can be made between the number of metamodel elements and relationships in AIMES and other EAs).

4.2. To make the approach accessible to people with few IT or modelling skills, the model is kept as simple as possible (excluding some optional parts that do not have to be used), with just four views that each containing only one central concept. The CEO was able to work with AIMES.

4.3. Throughout the different rounds of the research process, the researcher guided the CEO in the development of the EA model. The ultimate goal is to further develop the AIMES approach so that any need for external help is reduced to a minimum.

4.4. A process overview can be built with the business view. Processes (or projects) can be elaborated by using a business process management approach (or project management approach) and linking this to the corresponding process (or project) in the AIMES model.

4.5. The CEO was involved in developing the AIMES model, as he possessed the required knowledge to make an overview of the SME. The AIMES model is an instantiation of the AIMES metamodel that is developed and further refined throughout the research cycles, based on the problems the CEO and the researcher encountered.

4.6. In terms of complexity, the number of metamodel concepts and relationships of AIMES is considerably lower than in other EA frameworks and kept to the bare minimum.

The main benefits in the SME from the research were twofold.

i. The EA was built from scratch, this offered considerable insight into the structure and inner workings of the SME. It was clear that the CEO became very enthusiastic after he had explicated his goals for the SME, because he experienced this entire process as a steep learning curve.

iii. AIMES provided the SME a platform for analysis and guided change, especially because of the built-in traceability by integrating four viewpoints into one metamodel.

6.5.3 Research Evaluation Result

The research effort (integrated model) demonstrated that AIMES enables the development and management of an EA model for SMEs. It made the CEO think about his SME, how things work, why things are done, who is involved in and responsible for what, what the conflicting goals of different stakeholders are, and how balanced decisions should be made between these conflicting goals. In this respect, one specific advantage was that the CEO of the study SME became able to assess which operations could be executed by software instead of by the employees that executed them up to that moment.

In general, it is safe to say that the AIMES model enabled a better control of the SME, with improved communication and interaction, by offering a holistic overview, in which elements are part of a bigger picture. The approach was primarily used in a top-down manner (i.e. from business *strategy* to *Operations*), thus increasing the CEO's control of the SME. At the same time, though, AIMES also increased communication and interaction among employees and other stakeholders, as it was also used to discuss parts of the model with them. A final advantage could be that employees may become more motivated if they know how their role is situated within the bigger picture of the whole SME. This was not yet visible in the SME, but longer-term evaluation will undoubtedly provide more insight into this type of benefits. The evaluation confirms that AIMES metamodel supports the essential EA dimensions and meets the requirements for EA for SMEs.

6.6. Performance Evaluation

In order to measure the efficiency and performance of the integrated model for small and medium scale enterprise systems, the researcher carried out performance analysis of three enterprise architecture models including AIMES. The aim of the performance evaluation is to demonstrate that business system developed using AIMES model when compared to business systems developed using other enterprise model will possess the following attributes;

i. The system will be integrated with high degree of data integrity
ii. The system will support widely accepted architectural principle of;
    a. Data capture at source
    b. Data consistency and maintainability
    c. Data redundancy
    d. Separation of concerns
iii. The system will support fundamental business rules
iv. The system will have well defined functional scope and interface
v. The system will support data sharing, reduce data redundancy and have a high degree of data consistency.

Three enterprise architecture models including AIMES were evaluated. The models are; ARIS, CIMOSA and AIMES.

ARIS: Architecture for Information Systems

CIMOSA: Computer Integrated Manufacturing Open System Architecture

AIMES: An Integrated Model for Small and Medium-Scale Enterprise Systems

6.6.1 Method and Technique

In order to critically evaluate the efficiency and performance of the models with respect to the stated characteristics, a standard technique known as Figure-of-Merit Analysis was used with a set of four Architectural Evaluation Criteria.

6.6.2 Brief Definition of Figure-of-Merit (FoM) Analysis

The figure-of-merit data analysis technique is a mechanism used to perform and score the comparative analysis of the two or more business systems. The Figure-of-Merit data analysis technique has been taken from a Strategic Planning paper presented to the Australian Computer Society in 1988 (Ng, 1988) and the technique is used extensively in the Australian Department of Defence and has been used in more than twenty-six (26) corporations worldwide since 1988. It was first introduced into the RAAF in 1988 and has since been extended to the other services.

The steps performed in the Figure-of-Merit Analysis are:

i. Based on the Architectural Evaluation Criteria and their relative significance, assign a Weighting Factor (WF) to each of the Architectural Evaluation Criterion to reflect its relative significance. Weighting factors were assigned to the three enterprise architecture models

ii. Calculate the Unitizing Factor (UF) as follows:

$$UF = \frac{\text{Sum of WFs}}{\text{Number of WFs}} \quad \ldots\ldots\ldots\ldots(1)$$

iii. For each of the model, assign a raw score on a relative scale of 1 to 10 with respect to each of the Architectural Evaluation Criterion.

iv. Unitize the raw score as follows:

$$\text{Unitized Score (US)} = \frac{\text{Raw Score} * WF}{UF} \quad \ldots\ldots\ldots\ldots(2)$$

v. Calculate the Average Unitized Score (AUS) for each EA model:

$$AUS = \frac{\text{Sum of Unitized Scores (US)}}{\text{Number of Architectural Evaluation Criteria}} \quad \ldots\ldots\ldots\ldots(3)$$

The AUS is the ultimate indication of how well a Business System is rated against the Architectural Evaluation Criteria. The rounded off AUS is thus the Figure-of-Merit (FoM).

The researcher assigned the raw score to each of the Architectural Evaluation Criterion for the three models and followed all the steps to calculate the Figure-of-Merit for the three EA Models under assessment.

6.6.3. Brief Description of the Architectural Evaluation Criteria

The architectural evaluation criteria are described in detail below.

1. Architectural Principles

Architectural Principles refer to the good practices in the overall design of business systems. In the architectural evaluation of the two models designed to meet the needs of the business, the following architectural principles were used:

a. Data Captured at Source

Capturing data where it occurs, that is, at its source, is deemed to be a good system design practice. If the same kind of data is captured by different business functions and not centrally coordinated, duplicated capturing of the same data may result. This in turn leads to data inconsistency and data redundancy. The measure used for this criterion is to examine the technical documentation of the system to identify the business function that initially creates the definition of the data, that is, the source of data capture. In addition, the number of business functions that create the data provides useful information as to whether this criterion is observed in the system design.

b. Data Consistency and Data Maintainability (100% Principle)

The way the data is structured must be based on the inherent nature of the business. If the data is not structured properly it is impossible to maintain its accuracy. Hence, the data will become inconsistent and not maintainable. A well accepted good design practice called the 100% Principle, a fundamental architectural principle advocated by the International Standards Organisation (ISO) which stipulates that any business rule (which may be applicable to different business functions) must be asserted once and in one place only. This eliminates the possibility of the same business rule being asserted inconsistently in different business functions. When implemented in an information system, these business rules will be implemented as data integrity rules residing with the data to which they apply. The measures used for this criterion relate to the basis of the data structure, adherence to the 100% Principle, whether the same data are captured and stored more than once, and poor structuring of the data. The basis of the data structure should be based on the inherent nature of the business in terms of the business requirements and the business rules, with the outcome of a normalized data structure. The business rules may be specified as Structured English Sentences based on the data structure.

The 100% Principle advocates that all (i.e. 100%) of the rules concerning the updates of data be handled centrally by the database management system rather than by each and every one of the application programs that invoke the rules. Apart from eliminating duplicated efforts, it guarantees that the data is updated consistently across the board and renders the data more maintainable. The alternative to adherence to the 100% Principle is the hard coding of the business rules in each and every one of the application programs. Poor structuring of the data often brings about hard coding, but hard coding itself leads to poor data structuring over the maintenance life of the information system.

c. Data Redundancy

Data Redundancy results from the same data being captured and stored more than once or poor structuring of the data. Apart from wasting data storage it renders the data accuracy and quality not maintainable. The measures used for this criterion are the basis of the data structure, adherence to the 100% Principle, whether the same data are captured and stored more than once, and poor structuring of the data.

d. Separation of Concern

Another fundamental architectural principle is the principle of separation of concern. It stipulates that business concepts and business processes that are closely related should be grouped together and business functions should only be concerned with those business concepts and business processes that are relevant to it. This principle ensures that each business function, based on which a component is built, is highly cohesive. If a system is highly modularized, that is, broken down into program modules in a well-defined manner, then changing the logic of a program module may not affect other program modules. This can significantly reduce maintenance costs of the system.

The measures used for this criterion are the clustering of business entities into business themes and the hierarchical decomposition of business functions as the basis of well-defined program modules and system components

2. Fundamental Business Rules

Fundamental Business Rules are the rules under which the mainstay of the business operates upon which the data structure is based. If the data structure does not reflect these business rules it is impossible for the system to satisfy the business requirements.

The measure used for this criterion is the ability of the data structure, as defined in the Conceptual Schema (Logical Data Model), to support the defined business rules as described in the AIMES assumptions and business rules.

3. Functional Scope and Interfaces

Functional Scope refers to the totality of the functions performed by the system. The interfaces refer to the interfaces among these functions. This evaluation criteria looks at the coverage of the system in terms of the functions and their interrelationships.

The measure used for this criterion is the totality of the business functions performed by the system and their adherence to the hierarchical decomposition of business functions as described in the Enterprise Model.

4. Data Sharing, Data Redundancy and Data Consistency

Different functions in a system may use the same data. Sharing data across the functions is the basis for integrating different aspects of a system or different systems. If the data is shared properly then the benefit of an update to a piece of shared data is automatically shared across these systems. This eliminates the need for duplicating data and reduces the overall costs of the systems. The measure used for this criterion is the ability of different business functions in the system to use and share the same data as defined by the Enterprise Model, where the data modelling and data definition is based on conceptualization. The level of integration based on an Enterprise Model, ensures that no redundant data will be created and that the benefits of an update to the data will be shared across all information systems using the data.

6.7 Findings and Analysis

Using these four architectural evaluation criteria, the researcher allocated a ‘2’ to the Weighting Factor of the most important dimension, that is, Fundamental Business Rules.

We considered Functional Scope and Interfaces as the least important architectural evaluation criteria and allocated a ‘1.25’ to the Weighting Factor of this dimension. The researcher believes this was due to the two approaches using the same IT specification of requirements.

Data Sharing, Data Redundancy and Data Consistency were regarded as the second most important dimension and allocated a ‘1.75’ to its Weighting Factor relative to Fundamental Business Rules and Functional Scope and Interfaces.
The Architectural Principles as more important than Functional Scope and Interfaces as it pertains to the good design practices employed in developing an information system. However, we allocated a ‘1.50’ to the Weighting Factor of Architectural Principles as it was considered to be 75% relative to the Weighting Factor of Fundamental Business Rules.

These architectural evaluation criteria focus on the outcomes to be achieved in order to deliver an optimal information systems architecture. The objective is to evaluate the three enterprise architecture modes designed to meet the needs of the business, using different approaches, against the architectural evaluation criteria and the impact of implementing these business systems with respect to these criteria. It is important to note that these criteria are relevant in evaluating any information system. From the assignment above, we created Table 6.1;

Table 6.1. Measures Pertaining to Each Architectural Evaluation Criterion

S/N	**ARCHITECTURAL EVALUATION CRITERIA**	**WEIGHTING FACTOR (WF)**	**MEASURE**
1	Architectural Principle	**1.5**	1. Source of Data Capture
	a. Data Captured at source		2. Number of Business Functions That Capture

			Data
	b. Data Consistency and Data Maintainability (100% Principle)		3. Basis of Data Structure 4. Adherence to 100% Principe
	c. Data Redundancy		5. Same Data Captured & Stored more than once. 6. Poor structuring of Data
	d. Separation of Concern		7. Clustering of Business Entities into Business Themes 8. Hierarchical Decomposition of Business Functions 9. Well-defined Program Modules or System Components
2	Fundamental Business Rules	**2.0**	Ability of the Data Structure (Conceptual Schema) to support the defined Business Rules based on the Enterprise model
3	Functional Scope and Interfaces	**1.25**	The Totality of the Business functions performed by the System, as defined in the model
4	Data Sharing, Data Redundancy and Data Consistency	**1.75**	Ability of different functions in the System to Use and Share Data as defined in the enterprise Model

The weighting factors were assigned based on the order of importance of each architecture evaluation criteria.

6.8. Comparative Evaluation

The objective is to evaluate the three EA models designed to meet the needs of the business, using different approaches, against the architectural evaluation criteria and the impact of implementing these business systems with respect to these criteria. It is important to note that these criteria are relevant in evaluating any information system.

A summary of the case study comparative results of the three business systems is shown in Table 6.2. – EA models Comparative Results Summary Using Figure-of-Merit Analysis.

Based on the Figure-of-Merit analysis, using weighting factors and scoring out of ten, the relative strengths of the three models are reflected by their respective unitized scores as follows.

Table 6.2. EA Models Comparative Results Summary

S/N	Architectural Evaluation Criterion	Weighting Factor (WF)	Raw Score			WEIGHTED SCORE (WF *RAW SCORE)			UNITIZED SCORE(RAW SCORE *WF)/UF			AVERAGE UNITIZED SCORE SUM OF US/No Of Arch. Eva. Criteria = US/4		
			ARIS	CIMOSA	AIMES	ARIS	CIMOSA	AIMES	ARIS	CIMOSA	AIMES	ARIS	CIMOSA	AIMES
1	Architectural Principle	1.50	4.00	5.00	9.00	6.00	7.50	13.50	3.69	4.61	8.31	0.92	1.15	2.08
2	Fundamental Business Rules	2.00	2.00	4.00	10.00	4.00	8.00	20.00	2.46	4.92	12.31	0.62	1.23	3.08
3	Functional Scope & Interfaces	1.25	4.00	5.00	8.00	5.00	6.25	10.00	3.07	3.84	6.15	0.78	0.96	1.54
4	Data Sharing, Data Redundancy and Data Consistency	1.75	2.00	3.00	10.00	3.50	5.25	17.50	2.15	3.23	10.77	0.54	0.80	2.69
	Total		12.00	17.00	37.00	4.625	6.75	15.25	2.84	4.14	9.39	2.84	4.14	9.39

The Average Unitized Score (AUS) is the ultimate indication of how well a Business System is rated against the Architectural Evaluation Criteria. The rounded off AUS is thus the Figure-of-Merit (FoM). From the Table 6.2, we can deduce the following Based on the Figure-of-Merit analysis, using weighting factors and scoring out of ten, the relative strengths of the three EA models are reflected by their respective unitized scores as follows:

AIMES scored 9.39 out of ten, that is, 93.90% compliance with the Architectural Evaluation Criteria.
CIMOSA scored 4.14 out of ten, that is, 41.40% compliance with the Architectural Evaluation Criteria.
ARIS scored 2.84 out of ten, that is, 28.40% compliance with the Architectural Evaluation Criteria.

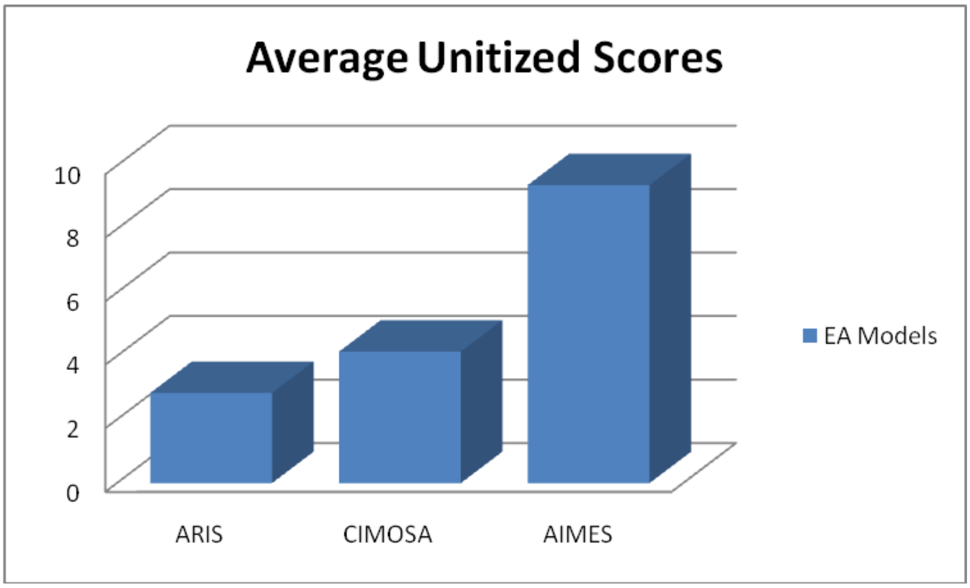
6.8.1. Summary of Evaluation Result

As can be deduced from the graphs below, the AIMES which scored at 93.90% compliance with the Architectural Evaluation Criteria meets all required business needs as expressed in its high score for Fundamental Business Rules and is architecturally sound with its flexibility to meet changing business needs in the future due to its compliance with the Architectural Principles. This result indicates that AIMES due to its flexibility will be less costly to maintain during its life than other models, such as the CIMOSA and ARIS, which are not architecturally sound.

On the other hand, it is clear that the ARIS which scored at 28.40% compliance with the Architectural Evaluation Criteria and CIMOSA which scored 41.40% compliance with the Architectural Evaluation criteria are not architecturally sound as it now stands, as they do not abide by the Architectural Principles and do not meet the business requirements as expressed in terms of the Business Rules. Hence, the ARIS and CIMOSA do not possess the flexibility to meet changing business needs in the future.

Table 6.3. AUS of the EA Models

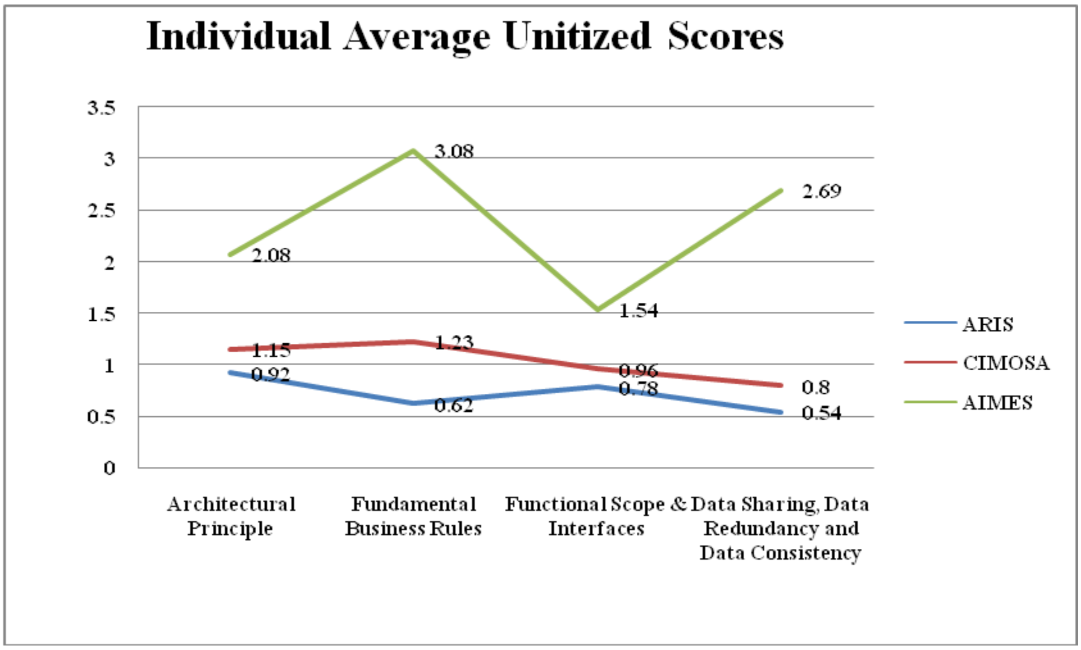
ARIS	**CIMOSA**	**AIMES**
2.84	4.14	9.39



Graph 6.1. Figure-of-Merit Analysis for the three EA Models using Total AUS

Table 6.4. Individual AUS of each Architectural Evaluation Criteria

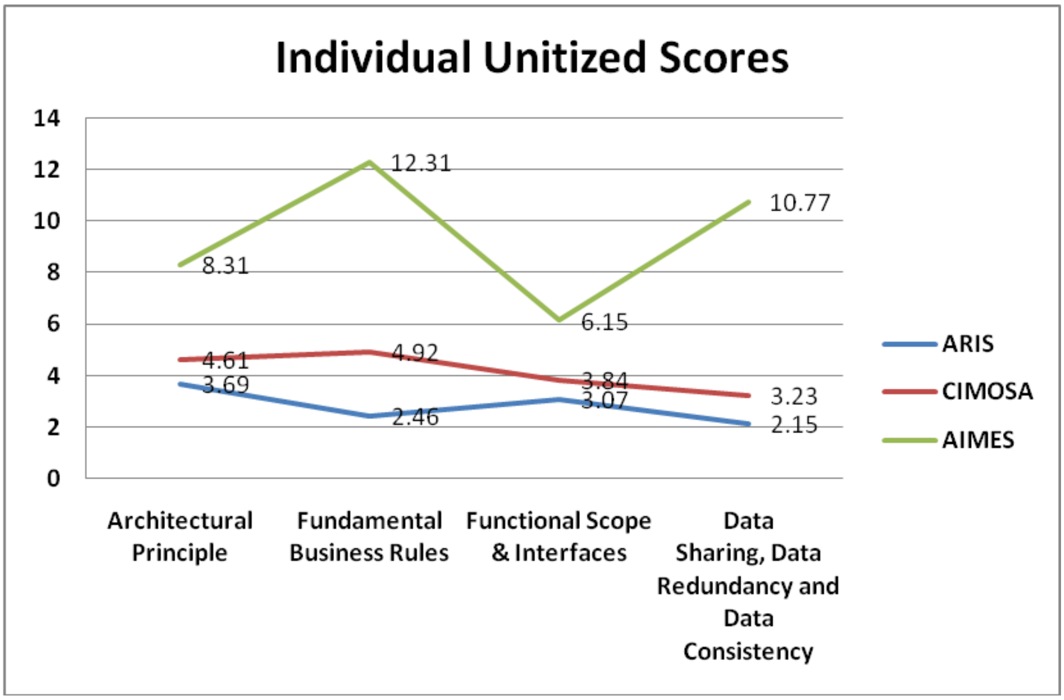
Architectural Eval. Crit	ARIS	CIMOSA	AIMES
Architectural Principle	0.92	1.15	2.08
Fundamental Business Rules	0.62	1.23	3.08
Functional Scope & Interfaces	0.78	0.96	1.54
Data Sharing, Data Redundancy and Data Consistency	0.54	0.8	2.69



Graph 6.2. Individual AUS of each business System w.r.t Architectural Evaluation Criteria

Table 6.5 Individual Unitised Score of each Business System w.r.t Architectural Evaluation Criteria

Architectural Eval. Crit	**ARIS**	**CIMOSA**	**AIMES**
Architectural Principle	3.69	4.61	8.31
Fundamental Business Rules	2.46	4.92	12.31
Functional Scope & Interfaces	3.07	3.84	6.15
Data Sharing, Data Redundancy and Data Consistency	2.15	3.23	10.77



Graph 6.3. Individual Unitized Scores w.r.t Architectural Evaluation Criteria

6.8.2. Evaluation Conclusion

The findings support the main attributes of this evaluation that an information system designed and implemented in accordance with the Integrated Model is superior to an information system that does not use an Integrated Model as its basis of development. The information system designed on the basis of an Integrated Model will be naturally integrated with the other information systems and its data will have a high degree of data integrity. Specifically, the findings support each of the following attributes:

i. The system will be integrated with high degree of data integrity
ii. The system will support widely accepted architectural principle of;
    a. Data capture at source
    b. Data consistency and maintainability
    c. Data redundancy
    d. Separation of concerns
iii. The system will support fundamental business rules
iv. The system will have well defined functional scope and interface
v. The system will support data sharing, reduce data redundancy and have a high degree of data consistency.

6.9 Framework for Systems Engineering Case Study Research Concept Domain for EA System Evaluation

The AIMES was further compared with existing systems based the Martin's (martin, 2006) framework for systems engineering case study research concept domain. The AIMES was developed using Enterprise Architecture approach and it is based on the requirements for EA for SMEs. EA is still unknown and hardly used in SMEs. The comparisons are as shown in table 6.6

Table 6.6 Enterprise Architecture Case Study Framework

Concept Domain	New AIMES	Existing Systems
A. Strategic Technical Planning	Yes	No
B. Capability – Based Analysis	Yes	No
C. Technology and Standard Planning	Yes	No

D. Enterprise Requirements Definition and Management	Yes	No
E. Enterprise Architecture and Conceptual Design	Yes	No
F. Program and Project Detailed Design and Implementation	Yes	No
G. Program Integration and Interfaces	Yes	No
H. Program Validation and Verification	Yes	Yes
I. Deployment and Post Deployment	Yes	Yes
J. Program Life cycle Support	Yes	No
K. Risk and Opportunity Management	Yes	No
L. Enterprise Evaluation and Assessment	Yes	No

The result shows that there is a huge difference between the AIMES system based on EA and existing systems as shown in table 4.

Table 6.7. Comparison of Existing Situation and New (Future) Situation

Existing System in the SME	**New AIMES System for the SME**
Pockets of Information System residing in Silos	Integrated information system – Eliminates stovepipes
Multiple Source of the same information	Provided a single Access point for users
The SME was unable to share information	Improved Access to accurate and timely information
Lack of standardization in Application Development	Single Application Development Environment
Applications are not Integrated	Ensures data quality providing a single source of truth – Integrated Application
Security Threats	Improved security – User accesses only data needed for his function
Increasing Complexity and technology choices for solutions	Use Current technology and reduce complexity
Incompatibilities between Hardware and Software platform	Enhanced Interoperability between hardware and Software platforms
Lack of IT-Business Alignment	Improved IT to Business Alignment
Rigid Inflexible technical Infrastructure	Current and Flexible technology Infrastructure

Rigid, brittle, aging systems	Based on business processes and uses technology enable business capabilities
Inconsistent, duplicated islands of data	Eliminated redundant data entry and improved information accuracy

6.10 Comparison of AIMES with Other Architecture Frameworks

The AIMES architecture was compared with some existing models/architecture based three (3) most recognized areas of enterprise architecture which are: (1) Life cycle (2) Model views and (3) Genericity Levels,

For the purpose of this comparison, we are going to use the short name (abbreviations) to represent the various architectures;

CIMOSA = Computer Integrated Manufacturing-Open System Architecture
ARIS = Architecture for Information Systems
TOGAF = ISO Reference Model of Open Distributed Processing
AIMES = An Integrated Model for Small and Medium-Scale Enterprise Systems

6.10.1 Comparison by life cycle

Table 6.8. Comparison by Life Cycle

Architecture	Identification	Concept	Requirement	Design	Implementation	Operation and Maintenance
CIMOSA	Not defined	Not defined	Requirement Definition	Design Specification	Implementation Description	Operation, Model maintenance
ARIS	Not defined	Not defined	Operation concept	IT system concept	implementation	Not defined
TOGAF	Not defined	conception	Requirement definition	IT Design Specification	Implementation	Evolution
AIMES	Enterprise	Conception	Requirement Definition	Design Specification	Implementation Description	Operation, Continuous Improvement , Dissolution

From Table 6.8, TOGAF cover the two uppermost layers of the enterprise life cycle, that is, the identification of the business entity and definition of concepts. This information is assumed to be provided by enterprise management in all other methodologies. The enterprise operation is defined in TOGAF and CIMOSA. Model maintenance is explicitly identified in CIMOSA and contained in the evolution layer of TOGAF. ARIS only support requirements, design and implementation aspects. In addition, ARIS cover the life cycle of information systems while AIMES covered all the life cycle.

6.10.2 Comparison by Model Views

Table 6.9 Comparison by Model views

Architecture	**Function**	**Information**	**Decision/Organisation**	**Structure/ Resource)**
ARIS	Function view (static) Control view (dynamic)	Data View	Organization view	Resource View
CIMOSA	Function view (static), function view dynamic)	Information View	Organisation View	Resource View
TOGAF	Information viewpoint	Computational viewpoint	Enterprise Viewpoint	Technology viewpoint
AIMES	Information View	Information view	Enterprise view	Infrastructure and Resource View

In Table (6.9),CIMOSA assumes one consistent enterprise model in which particular vies are provided for the user in the engineering environment to allow for model engineering on a particular aspect of the enterprise Operation. ARIS provides a similar approach, but has identified the control view for integrating the different views in to a common process model.. TOGAF does not define model views explicitly but provides viewpoints on a common model. AIMES provides all the views to cover the fundamental views of enterprise architecture.

6.10.3 Comparison by Genericity Levels

Table 6.10 Comparison by Genericity Level

Architecture	**ARIS**	**CIMOSA**	**AIMES**	**TOGAF**
Generic	Generic	Generic	Generic	Not defined
Partial	Reference Models	Partial	Reference model	Not defined
Particular	Particular	Particular	Particular	Not defined

In Table (6.10), except for TOGAF, which only provides a single task module, all the other methodologies have a rather populated generic level and provide sets of partial/reference models.

CHAPTER SEVEN
SUMMARY AND CONCLUSION

7.0 Introduction

This chapter first summarizes the findings of the literature review and the findings from the system analysis. It then addresses important areas of future research relevant to Enterprise Modelling, with particular emphasis on the uses of an Integrated Model.

7.1 Summary

This dissertation presents the design of the AIMES model as the first effort to develop an integrated EA approach specifically tailored to SMEs. The AIMES model is designed according to the requirements for EA in an SME context. This is achieved by means of research in one specific SME. The resulting metamodel is expressed as a UML class diagram, and extended with concept and relationship definitions in ArchiMate. As the case SME implemented certain changes according to the insights gained from the EA model, it was clear that the AIMES model was indeed very valuable and involves a constant trade-off between comprehensiveness and simplicity. In order to find the right balance, a set of EA frameworks used in business and academia was analyzed to capture the essential dimensions of EA approaches. This model is based on the essential views and dimensions of EA frameworks and is kept simple so that it may be applied in an SME context. The final AIMES model includes five essential views (i.e. business, information, information system, resource and infrastructure and security and governance), one for each most frequently used EA focus. Finally, the AIMES metamodel is evaluated according to the dimensions essential in EA and requirements for EA in an SME context. The model provides a holistic overview of the SME and thus incorporates the essential dimensions of existing EA approaches.

From a Scientific/Practical point of view, SMEs are important for economy. However, not all new SMEs make it through the first years. 70 percent survive at least 2 years, 50 percent at least 5 years, a third at least 10 years and only a quarter stay in business 15 years or more (Bureau of Labour Statistics, 2011). EA could help SMEs in overcoming the problems related to lack of structure and overview and increase survival rate. However, EA is still unknown and hardly used in SMEs.

Our approach differs from existing EA approaches in that the approach is specifically designed taking into account the characteristics of SMEs and their CEO. As Lankhorst (2009) mentioned, it is necessary to use an EA approach that is understood by all those involved from different domains. SMEs have characteristics, some which are indeed different from larger companies.

This study proposes architectural design as the prime means to align business transformation with IT development. This alignment is supported by a single integrated architecture called the Integrated Model for Small and Medium-Scale Enterprise Systems (AIMES) in which business architecture is related to IT architecture. The architectural design approach underlying AIMES allows us to assess the impact of new business models on the information systems supporting these models, and the other way around, to assess the impact of new technologies on the business models. AIMES has been applied successfully to a project which we described in this dissertation.

The AIMES model serves as a reference point between different organisations and enables them to understand each other's framework. AIMES is a natural and useful resultant of the implementation of new models which are designed to meet high customer requirements and to increase enterprise promptitude and improve its ability to respond to the requirements of dynamic and borderless market place as well as to facilitate the integration of the enterprise economy.

7.2 Areas of Application

AIMES architecture has a strong orientation towards its practical application to small and medium-scale enterprises. The metamodel and methods can be applied to small and medium enterprises (SME's) as well as holdings and virtual enterprises of different sectors. The objective is that it can be applied to both new SMEs (at the formation phase) and SMEs in the execution phase that want to improve their performance and be competitive. However, at the moment all the work has been oriented more on real SME. The areas are;

i. Manufacturing companies
ii. Educational institutions

iii. Banks and Financial institutions

iv. Transportation

iv. Construction companies

v. Information Technology etc.

7.3 Major Contribution to the Body of Knowledge

This research proposes an enterprise architecture (EA) approach that can be used by small and medium-scale enterprises (SMEs) to develop their EA models and manage their EA. The approach differs from existing EA approaches in that the approach is specifically designed taking into account the characteristics of SMEs and their CEO. A crucial element that was missing for SMEs to be able to use EA was simplicity in the existing approaches. The research among other things;

i. Created an enterprise architecture metamodel for the SME. This research developed the AIMES metamodel for EA in Small and Medium Enterprise Systems (SMEs) context, an artifact that can be further refined and tested in other areas. The metamodel expressed the components and semantics of an architectural model for process-oriented SMEs using five views that separately addresses different concerns related to the business, information, Information system, Infrastructure and resources and governance and security. These aspects were coherently integrated in the enterprise architecture model that aims fulfilling the purpose. The metamodel identifies a baseline set of physical entities, logical entities, and relationships among these entities so that an enterprise architecture model can be developed coherently to answer core business questions and achieve the purpose of the architecture. The motivation for this effort has been to provide value quickly with a modest investment of resources.

ii. Established enhanced methodology for enterprise architecture development in SMEs.

The enhanced methodology provides a step-wise prescriptive approach for developing enterprise architecture for SMEs, prescribing the steps to be taken to deliver a robust enterprise architecture. Towards this objective, this research identified a set of enterprise architecture practice that must be followed to establish enterprise architecture for SMEs.

This enterprise architecture method requires that the enterprise architecture effort ascertain:

- The purpose of the enterprise architecture
- The different views to be captured by the architecture.
- The services and capabilities needed to develop the architecture.
- The metamodel of the EA for SME
- The technical reference architecture. Infrastructure and resources capabilities, technology support models.
- Integrate the Architecture on key artifacts and services given and received.
- Enforce Architectural Control: Choose key technologies and standardize.
- Validate and execute the architecture.
- Identify the desired enhancement to business and then final dissolution where necessary.

iii. Apply the Integrated Model to an SME (Donavan Tiles): The integrated model facilitates traceability from business strategies to business operations as well as the underlying technical design and platform deployment of information system for effective business transformation. An actor plays a role, a system fulfils an application component, an application component accesses data, data component accesses infrastructure component (artifact) and infrastructure services made available to applications. Services provide a way to show the alignment between the different modelling layers. The integrated model shows the role of information system and technology in the SME in order to control its life cycle and also align information system with business process and information thus establishing a reference for efficient resource management.

iv. Build a Business Application classification model for SMEs based on the integrated model. The Information system view of the SME needs to be planned and managed. The application classification model provides instrument for balancing the application portfolio and life cycle of applications. Classifying applications is an effective way of managing applications portfolio. Towards this objective, this research built an application classification model for the SME that can be used as a tool in analyzing and enhancing the information system view development process. The purpose is to use it in analyzing initial information system view and in developing and describing target information system view.

v. Implemented an integrated enterprise information system for the SME based on the integrated model. Integrated information system is needed in the SME in order to eliminate stove-pipe applications, standardize the business processes and streamline operations. This will reduce cost of maintaining disparate systems and cost of technology. Towards this objective, this research applied the integrated model to an SME (Donavan Tiles) and developed an integrated enterprise information system for the SME. The information system is based on core business processes of the SME and uses technology enabled business capabilities. The Information system;

- Allows adequate support for business activities.
- Produces timely and accurate information.
- Allows easy adjustment to organization and business changes.
- Produces Overall cost minimization.

EA for SMEs could provide SMEs with solution to problem related to system complexity, IT- Business alignment and lack of structure and overview of business processes. The research also identified the criteria by which to judge the research and demonstrate how these criteria are met.

7.3.1 Research Communication

To communicate our work we have published four (4) articles:

i. Mgbeafulike Ike and Okonkwo Obikwelu R. (2013). Enterprise Architecture – A Tool for Business Innovation Realization in the Enterprise. The International Journal of Engineering and Science (IJES) (www.theijes.com) Volume 2 Issue 5 Pp. 26-29, 2013. It describes enterprise architecture as a good tool for realizing business value.

ii. Mgbeafulike Ike and Okonkwo Obikwelu, R. (2013). Architecture Framework for Resolution of System Complexity in an Enterprise. IOSR Journal of Computer Engineering (IOSR-JCE) (www.iosrjournals.org) Volume 10, Issue 4, PP 01-05, (Mar. - Apr. *2013*). It proposes the use of integrated EA model in resolving IT system complexity.

iii. Okonkwo Obikwelu, R. and Mgbeafulike Ike, J (2013). An Integrated Framework for Efficient Implementation of Enterprise Systems. International Journal of Engineering and Computer Science (www.ijecs.in) Volume 2 Issue 6 Page No. 2102-2111, June 2013. It proposes the principles, the motivations and the models of our proposed architecture.

iv. Mgbeafulike Ike J. and Okonkwo Obikwelu, R. (2015). Design of an Integrated Model for Development of Business and Enterprise Systems. International Journal of Research Studies in Computer Science and Engineering (IJRSCSE). (www.Arcjournals.org). Volume 2 Issue 5 Page No. 50-57, May 2015. It proposes the use of the integrated model for the design and development of business and information system.

7.4 Suggestions for Further Research

The researcher considered it far from saying the reality if it is assumed that this architecture is very efficient and has taken care of the entire problem associated with complex system development in SMEs and resolved the IT resource and business alignment. In as much as it has been able to achieve some relative success in enterprise architecture implementation and management, there are still more work to be done. The researcher therefore suggests that the AIMES architecture be improved to accommodate new developments in enterprise architecture such as extending it to carter for Virtual small and medium scale enterprises (AIMES architecture for virtual SME integration) – for the design of high performance internal and cross organisational business processes supported by the new information technologies (e-commerce, knowledge management, workflow etc). Another area of future research is in extending AIMES to build a Knowledge management model to facilitate enterprise architectural approach with an emphasis on strategic knowledge support. This will support knowledge discovery and facilitate the exploitation of information captured through the application of AIMES framework. Further research should also look at mathematical model for representation of enterprise architecture to show how each view is mathematically related to each others as a set of solvable equations.

7.5. Recommendations

This research work is recommended for small and medium-scale enterprises' (SMEs) and any company that wants to standardize their processes and streamline their internal operations. As a result of the need for efficiency, accuracy and timeliness, we recommend based on the research findings a research on the extension of AIMES to achieve Intelligent Enterprise Integration Architecture, Enterprise Migration and Virtual Enterprise Integration of enterprises in the value chain of SMEs.

A comprehensive training program and computer experts and operation in a network environment, the type of computers and allied communication hardware, software etc should be given a thorough consideration before implementation.

Finally, it is recommended that software architects can evaluate the architecture for the sole purpose of improving and formalizing it and adding new module which have been omitted for constraints of finance and time and pragmatic implementation of the project for profit maximization.

7.6. Conclusion

Enterprise architecture (EA) is a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and IT infrastructure. Recent research indicates the need for EA in small and medium-sized enterprises (SMEs), important drivers of the economy, as they struggle with problems related to a lack of structure and overview of their business. However, existing EA frameworks are perceived as too complex and, to date, none of the EA approaches are sufficiently adapted to the SME context. Therefore, this dissertation presents the AIMES model for EA in SMEs that was developed and evaluated through case study research in an SME. The research presents the design of the AIMES model as the first effort to develop an integrated EA approach specifically tailored to SMEs. The AIMES model is developed with special attention towards the characteristics and requirements for EA in an SME context. This is achieved by means of case study in a specific SME. The model consists of defining and understanding the different elements that shape an organisation and how these elements are inter-related. In this dissertation, we proposed a set of concepts and their relationships to describe the enterprise with the purpose of understanding and facilitating its evolution. These concepts are part of an enterprise architecture that is decomposed in five architectural views, each focusing on separate concern with the enterprise. The views as well as their interrelationships were formally defined through an integrated metamodel. Based on this metamodel, consistency between views is ensured through the definition of constraints relating model entities, belonging to the same (view consistency) or different (inter-view consistency) views. The small number of views enables the easy manipulation of them. With the integrated model, IT planning was improved, IT complexity and cost was reduced by establishing a process that is focused on building, maintaining, acquiring and retiring technology. The model also improved IT to business alignment by facilitating the adaptation of technology to changing business needs and pressure in the enterprise.

The methodology proposed addresses all issues related to enterprise information system design, utilizing the views defined, through eight discrete stages (Requirement Definition, business view design, information view design, services and system design, infrastructure and resources view design, enforce architecture control, validate and execute, evolve and continuously improve). Moreover, AIMES supports model exchangeability through the transformation of the common metamodel to internal tool-specific metamodel.

This research is claimed to be a significant and original contribution to the solution of increasingly important problem of lack of structure and overview in Small and Medium-scale Enterprises (SMEs) thereby bridging the gap between small and medium-scale enterprises (SMEs) and enterprise architecture (EA). The architecture is bound to change the fortunes of any SME towards effectiveness and profitability.

REFERENCES

Allen, B. R. & Boynton, A C. (1991). Information architecture: In search of efficient flexibility. *MIS Quarterly*. Vol. 15, No. 4. Pp. 435-444.

Ambler, S.W. (2002). Architecture and Architecture Modelling Technques. URL: http://www.agiledata.org/essays/enterprisearchitecturetechniques.html. Accessed 12-12-2-12.

AMICE (1993) *CIMOSA: Open System Architecture for CIM*, second edition, Springer-Verlag, Berlin

Anthony, R N. (1965). Planning and Control: A framework for analysis. Cambridge, MA, Harvard university Press.

Aarabi, M., Saman, M. Z. M., Wong, K. Y., Beheshti, H. M., & Hemdi, A. R. (2011). The Effect of Enterprise Architecture for Enterprise Resource Planning in Small and Medium Enterprises: a Review and Research Direction. In IESS, Indonesia, (pp. 159-163).

Ballantine, J.A., Galliers, R.D & Stray, S.J (1994). Information systems/technology investment decisions: The use of capital Investment appraisal technique in organisations. In: Proceedings of the 1st European Conference on IT Investment Evaluation. Henley, England, September, pp. 148-166.

Balabko, P., & Wegmann, A. (2006). Systemic classification of concern-based design methods in the context of enterprise architecture. Information System Front, Vol.8, No.2, pp.115-131.

Bernaert, M., Poels, G., Snoeck, M., & De Backer, M. (2013). Enterprise Architecture for Small and Medium-Sized Enterprises: A Starting Point for Bringing EA to SMEs, Based on Adoption Models. In Information systems and small and medium-sized enterprises (SMEs): state of art of IS research in SMEs: Springer.

Benbasat, I., Dexter, A S., Drury, D H. & Goldstein, R C. (1984). A Critique of stage Hypothesis Theory and Empirical evidence. *Communications of the ACM*, Vol. 27, No. 5, Pp. 476-485.

Benjamin, R I., Rockhard, J F., Scott Morton, M S. & Wyman, J. (1984). Information technology: A strategic opportunity. *Sloan Management Review*. pp. 3-10.

Bidan, M., Rowe, F., & Truex, D. (2012). An empirical study of IS architectures in French SMEs: integration approaches. European Journal of Information System, Vol. 21 No.3, pp. 287-302.

Beveridge, T. & Perks, C. (2000). Blueprint for the flexible Enterprise. Intelligent Enterprise. Vol.3 No. 4. Pp. 1-10.

Booch, G. (2005). "The Complexity of Programming Models". Keynote talk at AOSD, Chicago, Il.

Brancheau, J.C, Schuster, L & March, S.T (1989). Building and Implementing an Information Archiecture. Database Vol. 20, No. 2, Summer, Pp. 9 – 17.

Braun, C., & Winter, R. A. (2005). Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform. In EMISA, 2005, pp. 64-79.

Broadbent, M. & Weill, P. (1993). Improving business and information strategy alignment: Learning from the banking industry. *IBM Systems Journal*, Vol.32, No. 1. Pp. 162-181.

Browne, G.J & Rogich, M.B. (2001). An Emperical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques. Journal of Management Information Systems. Spring 2001. Vol. 17, No. 4.pp. 223-249.

Brosey, W. D., Neal, R. E., & Marks, D. F. (2001). Grand Challenges of Enterprise Integration 8th IEEE International Conference on Emerging Technologies and Factory Automation, 2, 2, (pp. 221 -227).

Buchanan, R.D. (2001). Assesing Enterprise Architecture Program Value. META Group Report 128, July 3. The Meta Group, 208 Harbor Drive, Stanford, CT06912-0061.

Bureau of Labor Statistics US (2011). Business Employment Dynamics. U.S. Department of Labor. http://www.bls.gov/bdm/. Accessed September, 12, 2015.

Bush, Vannevar (1945), "As We May Think," *The Atlantic Monthly*, Vol. 176, No 1, pp 101-108.

C4ISR (1997) - Department of Defense Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance Architecture Framework, version 2.0.

Capgemini, (2007). Service–Oriented Architecture the way we see it. URL: http://www.capgemini.com/ Accessed 12/12/2012

Card D. N. & R. L. Glass (1990). Measuring Software Design Quality, Englewood Cliffs, NJ:Prentice Hall.

Central Bank of Nigeria (CBN) (2001), First Annual Monetary Policy Conference on Growing the Nigerian Economy.

Chan, Y.E., Huff, S.L., Barclay, D.W. & Copeland, D.G. (1997). Business Strategic Orientation, Information Systems Strategic Orientation, and Strategic Alignment, *Information Systems Research* **8**(2): 125–150.

Chan, Y.E. (2002). Why Haven't we Mastered Alignment?: The importance of the informal organization structure, *MIS Quarterly Executive* **1**(2): 97–112.

Chan, J.O. (2008). An Integrated Architecture for Enterprise Relations Management. Communication of International Information Management Association, 8(2), 55-66.

Chen, P.(1976). The Entity Relationship Model – Towards a unifed view of data. *ACM Transactions on Database Systems.* Vol. 1, No, 1. Pp. 9-36.

CIO Council (1999). Federal Enterprise Architecture Framework (FEAF), Version 1.1,. URL: http://www.cio.gov/documents/fedarch1.pdf, Accessed 28.10.2012.

CIO Council (2001). A Practical Guide to Federal Enterprise Architecture. Chief Information Officer Council Version 1.0.

Clemons , E.K. (1991). Investments in Information Technology. Communications of the ACM. June. Vol. 34, No. 1. Pp.23-36.

Coad P. & E. Yourdon (1991). Object-Oriented Design, Englewood Cliffs, NJ: Prentice Hall.

Cook, M. (1996). Building Enterprise Information Architecture: Re-engineering Information Systems. 1.Edition. New Jersey. Prentice Hall PTR. 224 pages.

Davern, M.J & Kauffman R.J. (2000). Discovering Potential and Realizing Value from Information Technology Investment. Journal of Management information systems. Vol. 16, No. 4. Pp. 121-143.

Davenport, Thomas, H & Short, James E. (1990). “The New Industrial Engineering: Information Technology and Business Process Redesign”. Sloan Management Review. Pp 11 – 27.

Davenport, T.H. (1998). Putting the Enterprise into the Enterprise. Harvard Business Review. Vol.76, No. 4. Pp 121-131.

Davis G.B. (1982). Strategies for Information requirements determination. IBM Systems Journal. Vol.21, No. 1. Pp.4-30.

de Villiers, D.J. (2001). Using the Zachman Framework to Assess the Rational Unified Process, the Rational Edge. url: http://www.therationaledge.com/content/mar_0/t_zachman_dv.html.accesed 24.04.2012.

Devos J. (2011). IT Governance for SMEs. University of Ghent, Ghent

DoD Architecture Framework Working Group (2004). DoD Architecture Framework.

Dougherty D. (1992). A Practice-Centered Model of Organizational Renewal through Product Innovation. Strategic Management Journal Vol.13 No.S1: pp. 77-92.

Doumeingts G., Vallespir B., Zanettin M., Chen D. (1992) GIM: GRAI Integrated Methodology. A methodology for designing CIM systems. GRAI/LAP. Université-Bordeaux 1, versión 1.0.

Doumeingts G., Vallespir B., Chen D. (1993). Methodologies for Designing CIM systems. A Survey. Computers in Industry. 25:263-280.

Drury, D.H. (1983) An Empirical Assessment of the Stages of DP Growth. MIS Quarterly. Vol.7, No.5. Pp. 59-70.

Duhan S , Levy M & Powell P. (2001). Information Systems Strategies in Knowledge-based SMEs: The Role of Core Competencies; European Journal of Information Systems.

Enterprise Java Application Integration (eJai). (2001). *eJai Universal adapter Whitepaper*. Retrieved April 14, 2012 from the eJai Web site: http://www.igs.com/products/eJai%20Universal%20Adapter/eUA%20product%

European Commission (2003) Recommendation 2003/361/EC: SME Definition. Official Journal of the European Union 46 (L 124) (6)

European Commission (2011). Are EU SMEs recovering from the crisis? Annual Report on EU Small and Medium sized Enterprises 2010/2011.

Everden, R. (1996). The Information Framework. IBM Systems Journal. Vol. 35, No.1, pp37-68.

Evernden, R & Evernden E. (2003). “ Third Generation Information Architecture”. Communications of the ACM. Pp. 95-98.

Farbey, D.F & Willcocks, L.P (1998). Core IS capabilities for Exploiting information technology. MIT Sloan Management Review. Vol 39. No3. Pp. 9-21.

Farnum, C. (2002) "Information Architecture: Five Things Information Managers Need to Know," *Information Management Journal* Vol.36, No.5, pp 33-39.

Farrel D. (2003). The Real New Economy. Harvard Business Review. Pp.104-112.

Finkelstein, C. (1992). Strategic Systems Development. Singapore, Addison-Wesley Publishing Company.

Friedman, A. (1994). The stages model and the phases of the IS field. *Journal of Information Technology:* Vol. 9, pp 137-148.

Gable, J. (2002). Enterprise Application Integration. Information Management Journal. March/April Vol 36. No 2. Pp. 48-52.

Galliers, R D. & Somogyi, E K. (1987). From data processing to strategic information systems. A historical perspective. In Galliers, R D. & Somogyi, E K. eds. *Towards Strategic Information Sytems*. Abacus press. Pp 5-25.

Galliers, RD. (1987). Information technology planning within the corporate planning process. In Duling, A. & Berry, D. *Controlling projects within an integrated management framework*. Maidenhead, UK, pergamon Infotech State of the Art Report.

Gartner (2012). Gartner's 2011 Global Enterprise Architecture Survey: EA Frameworks Are Still Homemade and Hybrid.

Gibson , C. F. & Nolan, R L. (1974). Managing the four stagies of EDP growth. *Harvard Business Review.* January – February 1974. Pp. 76-88.

Gibson, R. (1994). "Global Information Technology Architectures," *Journal of Global Information Management* Vol.2, No.1, 1994, pp 28-38.

Goodhue, D. L., Quillard, J.A. & Rockart, J.F (1988) Managing the Data Resource: A Contingency Perspective. MIS Quarterly. Vol.12, No.3. Pp. 373-392.

Gorry, A.G & Scott Morton, M.S (1971). A Framework for Management Information System. Sloan Management Review. Autumn 1971, Pp. 55-70.

Gosling, D.H. (1993). Data Architecture: A blueprint for managing your business. CMA. The Society of Management Accountants of Canada. Hamilto. October. Vol 67. No 8. Pp. 17-23.

Greenhorst D. & Proper E. (2011). Architecture Principles – The Cornerstone of Enterprise Architecture. Springer.

Greenberger, Martin (1964), "The Computers of Tomorrow," *The Atlantic Monthly,* Vol. 213, No. 5, pp. 63-67.

Hambrick, D.C. & Mason, P.A. (1984). Upper echelons: The organization as a reflection of its top managers. Academy of Management Review, Vol.9, No.2, pp.193-106

Hackathorn, R.D. & Karimi, J.(1988) "A Framework for Comparing Information Engineering Methods," MIS Quarterly, 12:2, Pp. 203-220.

Harmon, P. (2003). Developing an Enterprise Architecture. Whitepaper available at the website of Enterprise Architect Magazine. URL:http://www.ftponline.com/ea/magazine/summer/online/pharmon_07_10_03. Accessed 24/04/2013.

Henderson, J. C. & Venkatraman, N. (1993), "Strategic alignment: Leveraging information technology for transforming organizations", IBM Systems Journal, Vol. 32, No.1, Pp.4-16.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information system research. MIS Quarterly, 28 (1), 71-107.

Heyse M, Bernaert M, Poels G (2012) Keuzes Maken binnen Processen: Het Vermijden van een Russische Roulette voor de Organisaties. University of Ghent, Ghent.

Hirschheim, R., Earl, M.J., Feeny, D. & Lockett, M. (1988). An exploration into the management of the IS function: Key issues and an evolving model. In Proceedings of the Joint International Symposium on IS: Information technology management for productivity and strategic advantage. IFIP TC-8. Open Conference Singapore.

Hoogervorst, J. (2004). Enterprise Architecture: Enabling Integration, Agility and Change. International Journal of Cooperate Information Systems, Vol. 13, No.3, pp. 213-233.

Huff B.L, Presley, A.R. & Liles D.H. (1993). A Comprehensive Enterprise Model for Small Manufacturers. In Proceedings of the 2nd Industrial Engineering Research Conference, 430-434. Institute of Industrial Engineers, Los Angeles, CA.

Hugoson, M., Magoulas, T., & Pessi, K. (2008) "Interoperability Strategies for Business Agility", Lecture Notes in Business Information Processing, Advances in Enterprise Engineering I, 4th International Workshop EOMAS, held at CAISE, Vol. 10; pp. 108-121.

IBM Institute for Business Value (2005). Component Business Models: Making Specialization Real 17p. (Online). Available: Http://www.G35.Ibm.com/services/us/imc/pdf/g510-6163-component-business-models.pdf. Accessed 20 October, 2015.

INCOSE (2005). *Systems Engineering Technical Vision, Version 1.0*, International Council on Systems Engineering.

Information Systems Architecture", IBM Systems Journal, Vol. 31, No. 3, pp. 590-616

Inmon, W.h. (1999). A brief history of Integration. eAI journal . http://www.eaijournal.com.

Jacobs, D., Kotzé, P., van der Merwe, A., & Gerber, A.(2011) Enterprise Architecture for Small and Medium Enterprise Growth. In EEWC (Vol. 79, pp. 61-75, LNBIP): Springer.

Johnson, P & Ekstedt M. (2007).Enterprise Architecture Models and Analysis for Information systems Decision Making. ISBN: 13-9789144027-524.

Jonkers, H., Lankhorst M., Van Brunen R., Hoppenbrouner, S., Bonsangue, M. & Vandertorre, L. (2004). Concepts for Modelling Enterprise Architectures. International Journal of Cooperative Information Systems. Vol. 13, No. 3.

Jonkers, H., Lankhorst, M. M., ter Doest, H. W. L., Arbab, F., Bosma, H., & Wieringa, R. J. (2006). Enterprise architecture: Management tool and blueprint for the organisation. Information System Front, Vol. 8 No.2, pp. 63-66.

Kamsties E, Hörmann K, Schlich M (1998) Requirements Engineering in Small and Medium Enterprises. Requirement Engineering Vol.3 No.2: pp.84-90.

Kalakota, R. & Robinson, M. (2000). E-Business 2.0. *Addison Wesley Longman, Reading*. p544.

Katzy, B.R., Eversheim W., Dobberstein,M., Feuerborn, K., Fiedrich,J., Krah, O., Muller, G., & Stepprath,F.J., (1993). CIMOSA Pilot Implementation for Technology Transfer. JSPE-IFIP WG. 5.3 Workshop on the Design of Information Infrastructure for Manufacturing (University of Tokyo, Japan), Pp. 51 -62.

Kemal A. Delic (2002).Enterprise It Complexity Ubiquity Magazine ACM New York, NY, USA Volume 2002 Issue, Article No. 8.

King, J L. & Kramer, K L. (1984). Evolution and organizational information systems. An assessment of Nolan's stage model. *Communications of the ACM,* Vol. 27, No. 5. Pp. 466-475.

Kumar & Van Hillegersberg (2000): Kumar, Kuldeep & Van Hillegersberg, Jos: ERP experiences and evolution. Communications of the ACM. Vol. 43 Issue 4, Pp. 22-26.

Kung, C.H & Solvberg, A. (1986). Activity Modeling and Behaviour Modeling In: Olle, T.W, Sol, H.G and VerrijnStaut, A.A.(EDs). Information Systems Design Methodologies improving the practice. Amsterdam. North-Hollangd IFIP,. Pp. 145-171.

Kramp T & Coulson, G. (2000). The Designof a flexible Communications Framwork for Next Generation Middleware. 2nd International Symposium on Distributed Objects and Applications, Proceedings, Pamela Drew, Robert Meersman, Zahir Tars, Roberto Zicari, Los Alamitos. IEEE Computer Society. ISBN: 0-7695-0819-7 Pp. 273 – 282.

Kroon, B., Voorde, K., & Timmers, J. (2012). High performance work practices in small firms: a resource poverty and strategic decision-making perspective. Small Bus Econ, pp.1-21.

Leck, C. (1997). Information systems frameworks and strategy. *Industrial Management and Data Systems*. No. 3, pp, 86-89.

Lankhorst, M. (2005) Enterprise Architecture at work. Modeling, communication, and analysis. Pp. 8. Berlin, Germany: Springer-Verlag Berlin Heidelberg.

Lankhorst, M. (2013). Enterprise Architecture at Work: Modelling, Communication and Analysis (Vol. 3). New York: Springer.

Lawrence, P.R & Lorsch, J.W (1968). Organisation and Environment, Boston Mass. Division of Research, Graduate School of Business Administration, Harvard University.

Lederer A.L & Mendelow A.L (1993). Information Systems planning and the challenge of shifting priorities. Information and management . vol 24. No.6. Pp.319-328.

Levy M. & Powell P. (2005) Strategies for Growth in SMEs: The Role of Information and Information Systems. Elsevier, Butterworth-Heinemann.

Levy M., Powell P., & Yetton P. (2001) SMEs: Aligning IS and the Strategic Context. Journal of Information Technolology Vol.16. No.3. Pp. 133-144.

Leifer, T.L, Lee, S & Durgee, J. (1994). Deep Structures, real information requirements determination. Information and Management. Vol. 27, No.5.Pp. 21-42.

Light, B., Holland, C.P & Wills, k. (2001). ERP and Best of breeds: A Comparative Analysis. Business Process management Journal. Vol. 7, No.3. Pp. 216 -224.

Linthicum, D.S (2001). Binding Companies before you take on B2B Integrator: Understanding Supply Chain Management. eAI Journal.

Linthicum D.S (1999). Enterprise Application Integration. Massachussets, USA, Addison-Wesley.

Lloyd, A, Dewar, R. & Pooley, R. (1999). Legacy Information Systems and Business process change: a pattern perspective. Communications of the Association of Information Systems. Vol 2. No. 24. Pp. 1-40.

Luftman, J. (2000). Assessing Business – IT Alignment Maturity. Communications of AIS. Vol.4. Article 14. Pp. 1-25.

Luftman, J. & Brier, T. (1999). Achieving and sustaining business-IT alignment. *California Management review*. Fall, pp. 109-122.

Luftman, J., & Ben-Zvi, T. (2011). Key Issues for IT Executives 2011: Cautious Optimism in Uncertain Economic Times. MIS Q Exec, 10(4), 203-212.

Lybaert, N. (1998). The Information Use in a SME: Its Importance and Some Elements of Influence. Small Business Economics, Vol.10 No.2, pp.171-191.

Lyytinen k & Hirschheim, R. (1987). Information Systems failure. A survey and classification of the empirical literature. Oxford Surveys in Information Technology. Vol 4. Pp. 257-309.

March, S. T., & Smith, G. (1995). Design and natural science research on information technology. Decision Support Systems, 15 (4), 251-266.

Martin, S. (1989). Information Engineering Book III. Englewood Cliffs, N.Y. Prentice-Hall.

Martin, J. N. (2006). An Enterprise Architecture Process Incorporating Knowledge Modeling Methods, Doctoral Dissertation, George Mason, University.

Mayer, R.J & Painter, M.K (1991). Roadmap for enterprise integration. In: Proceedings of Autofact’91 Conference, Chicago. Pp. 7.1 – 7.26.

McCarty, F., (1993). World-class manufacturing: a practical viewpoint. In: Proceedings of the Flexible Automation and Information Management, in FAIM, Limerick.

Mayer, R.J., Painter, M.K., (1991). Roadmap for enterprise integration. In: Proceedings of Autofact'91 Conference, Chicago, pp. 7.1-7.26.

McFarlan F.W (1984). Information Technology Changes the way you Compete. Harvard Business Review May –June, 62(3).1984. Pp. 93 – 103.

Microsoft Corporation. (2003). Microsoft Solutions Framework version 3.0. overview White Paper. Published. URL: www.microsoft.com. Accessed 1.3.2012.

Miller, E. (1999). Integrating PDM and ERP. Computer Aided Engineering. Vol.18. No. 3. Pp.69.

Mooney, J. Gubaxani, V & Kraemer, K. (1995). A process oriented framework for assessing the business value of information technology. Center for Research on

Information Technology and Organisations. IT in Business. University of California.

Mordi, F. (2005), SMEDAN Promises MSME Operators a Better Deal, Businessday, Businessday Media Ltd, Lagos

Morganwalp, J.M. (2003). A System of systems – Focused Enterprise Architectural Framework. Dissertation. George Mason University. 256p.

Morganwalp, J.M & Sage, A.P. (2003). A system of systems focused enterprise architecture framework and an associated architecture development process. Information knowledge and systems management. Vol. 3. Pp. 87-105.

Munro, M C. & Huff, Sl. (1985). Information technology and corporate strategy. *Business Quarterly*. Summer. Vol. 50, No. 2. Pp. 18-25.

National Research Council de EE.UU. (1986). Impact of integration efforts at McDonnell Aircraft Co., Deere and Co. Westinghouse Defense and Electronic Center, General Motors, and Ingersoll Milling Machine Co. Special Report to Nasa, Washingtong, DC.

Nell J.G. & Kurt Kosanke (1997). ICEIMT'97 International Conference on Enterprise Integration Modeling Technology. Accessed 07 Jan 2014.

Neuscheler, F & Spark D. (1994). The Economic View: A Concept Using Benchmarks to Analyze, Evaluate and Optimize business Processes. Proceedings of the IMSE 1994, Conference Grenoble.

NCR Corporation (1996). NCR Enterprise Architecture Framework. (www-document). NCR Corporation. URl:http://www3.ncr.com/architecture/togaf/print.htm. accessed 24.2.2012.

Ng, M. W. (1988) Project Evaluation and Selection in Information System Strategic Planning. *Proceedings of the Australian Computer Society Annual Conference.* Melbourne.

Niederman, f., Brancheau, J.C & Wetherbe, J.C (1991). Information Systems management issues for the 1990s. MIS Quatrely. December. Vol. 15, No. 4. Pp. 474-500.

Nolan, R L. (1979). Managing the crisi in data processing. *Harvard Business Review*. March – April. pp.115-126.

Notowidigdo, M H. (1984). Information systems: Weapons to gain the competitive edge. *Financial Executive*. Vol. 52, No 3. Pp. 20-25.

Omerzel, D.G. & Antoncic, B. (2008) "Critical Entrepreneur Knowledge Dimensions for the SME Performance", Industrial Management & Data Systems, Vol. 108, No. 9, pp. 1182-1199.

Oracle (2009). An Oracle White Paper in Enterprise Architecture, October.

Parker M.M & Benson R.J (1992). Information Economics. New Jersey, Prentice-Hall.

Periasamy, K P & Feeny, D, (1997). Information architecture practice: research-based recommendations for the practitioners, *Journal of Information Technology*, Vol, 12. Pp 197-205.

Pervan, G. (1998) "How Chief Executive Officers in Large Organizations View the Management of Their Information Systems," *Journal of Information Technology* Vol.13, No.2, pp 95-109.

Pessi, S. A. (2010) "An Introduction to Enterprise Architecture", 2nd edition, AuthorHouse, Bloomington, IN

Petrie, Charles J. (1992). Enterprise Integration Modeling: Proceedings of the First International Conference. MIT Press. ISBN 0-262-66080-6

Plachy, E.C & Hausler, P.A. (1999). Enterprise Solutions Structure. IBM Systems Journal. Vol. 38, No.1. pp 4-11.

Porter, M. (2001). Strategy and the Internet. Harvard Business Review. March 2001. Pp 63-78

Presley, A., Huff,A., et al (1993). A comprehensive Enterprise Model for Small Manufacturers. 2nd Industrial Engineering Research Conference. Los Aneles, CA.

Radeke, F.(2011). Toward Understanding Enterprise Architecture Management's Role in Strategic Change: Antecedents, Processes, Outcomes. In WI, (Vol. 2011).

Raghu-Nathan, B., Ragu-Nathan, T.S., Tu, Q., & Shi, Z. (2001) "Information Management (IM) Strategy: The Construct and Its Measure," *Journal of Strategic Information Systems* Vol.10, No.4, pp 265-289.

Regnell, B., Host, M., Dag, J.N., Beremark, P & Hjelm, T. (2001). An Industrial case study on distributed prioritization in market-driven requirements engineering for packaged software. Requirement Engineering. Vol. 6., no1. Pp. 51-62.

Reich, B.H & Benbasat, I (2000). Factors that Influence the Social Dimension of Alignment between Business and Information Technology Objectives. MIS Quarterly. Vol. 24, No. 1. Pp. 81 – 113.

Richardson, L., Jackson, B.M., & Dickson, G. (1990). A Principle-Based Enterprise Architecture: Lessons from Texaco and Star Enterprise. MIS Quarterly. Volume 14, Number 4. pp. 385-403

Riihimaa, J. & Ruohonen, M, (2002). Sahkoisesta kaupasta osaamisliiketoimintaan. Helsinki. MET Julkaisuja. In Finnish.

Ross, J W. & Weill, P.(2002). Six decisions your IT people shouldn't make. *Harvard Business Review*. Boston. Nov, Vol. 80, No. 11 pp, 84-96.

Ross J W., Weill P, Robertson D (2006). Enterprise Architecture as Strategy: Creating a Foundation for Business Execution. Harvard Business School Press, Boston.

Ross, J. (2003). Creating a strategic IT architecture competency: Learning in stages. MISQ Executive. Cambridge MA: Center for Information Systems Research. Volume 2, Number 1. pp. 31-43.

Ruh w, Maginnis, F. and Brown, W. 2000. Enterprise Application Integration. A Wiley Tech Brief. New York, USA. John Wiley and Sons. Inc.

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

Scheer, August-Wilhelm (1996) "ARIS-Toolset:Von Forschungs-Prototypen zum Produkt" Informatik-Spektrum 19: 71–78 © Springer-Verlag 1996.

Scheer, A.W, (1989). Enterprise-wide Data modeling. Berlin, Springer – Verlag.

Schekkerman, J. (2003). How to survive in the jungle of enterprise architecture frameworks: Creating or choosing an enterprise architecture framework. British Columbia: Canada: Trafford Publishing.

Schekkerman, J. (2003a). The Extended Enterprise Architecture Framework. Institute for Enterprise Architecture Development (IFEAD) URL: http://www.enterprise-architecture.info.

Schekkerman, J. (2003b). Enterprise Architecture Validation. September. Institute for Enterprise Architecture Development (IFEAD) URL: http://www.enterprise-architecture.info.

Schultz, M (2007). Architecture Principles: Creating the Foundation for Robust Architecture. IBM developerworks. (online). Available: http://www.ibm.com/developerworks/library/ar-archprn/. Accessed 20 October, 2015.

Sessions, R. MSDN homepage (2007). A comparison of the Top Four Enterprise-Architecture Methodologies [Learning Object] Available: http://msdn.microsoft.com/en-us/ library/bb466232.aspx#eacompar_topic8 (2011-04-18).

Simon, H.A (1960). The New Science of Management Decision. New York. Haper & Row. Pp 5-6.

Siau, k. (1999). Information Modelling and Method Engineering. A Psychological Perspective. Journal of database management. Vol.10, No.4, pp. 44-50.

Small Business Administration (2011). How Important are Small Businesses to the U.S. Economy? http://www.sba.gov/advo

Smith, J. A. (1998). Strategies for Start-Ups. Long Range Plan, Vol.31No.6, pp. 857-872.

Soliman, F., Clegg, S. & Tantoush, T. (2001), "Critical success factors for integration of CAD/CAM systems with ERP systems", International Journal of Operations & Production Management, Vol. 21 Nos 5/6, pp. 609-29.

Sowa, J F, & Zachman, J A, (1992). Extending and Formalizing the Framework for Information Systems Architecture. *IBM Systems Journal*. Vol. 31, No. 3. Pp. 590-617.

Spewak, S. (2000). Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology. John Wiley & Sons, New York, NY.

Sullivan, C H. (1985). Systems Planning in the information age. *Sloan Management Review*. Winter. Pp. 3-11.

Tamm, T., Seddon, P. B., Shanks, G., & Reynolds, P. (2011). How Does Enterprise Architecture Add Value to Organisations? CAIS, 28(1).

The Open Group (2004). URL:http://www.opengroup.org/ Accessed 22.4. 2012.

The Open Group (2002). TOGAF 8, The Open Group Architecture framework "Enterprise Edition", URL:http://www.opengroup.org/architecture/togaf. accessed 28.10.2011.

The Open Group (2009) TOGAF; version 9 Enterprise Edition. (Online). Available: http://pubs.opengroup.org/architecture/togafa.doc/arch/. Accessed 10 November, 2015.

Themistocleous, M & Chen, H (2004) Investigating the integration of SMEs' information systems: an exploratory case study. International Journal of Information Technology and Management 3(2/3/4), 208–234.

Themistocleous, M & Irani, Z. (2002). Evaluating and adopting Application Integration ; The Case of Multinational Petroleum Company, Proceedings of the 35th Hawaii International Conference of System Sciences.

Traverse, R. (2001), "An Intelligent agent framework for Enterprise Integration", IEEE transactions on systems, man, and cybernetics, Vol. 21, No. 6, pp. 1391- 1407.

Turban, E., McLean, E. & Wetherbe, J. (2001). Information Technology for Management Making Connections for Strategic Advantage. New York, Wiley 7 Sons. 791 p.

Umar, A., Karabatis, G.,Ness, L.Horowitz, B & Elmagardmid, A. (1999). Enterprise Data Qaulity: A pragmatic Approach. Information Systems Frontiers. Vol 1, No. 3. Pp 279-301.

Van den Hoven J. (2003). Data architecture: Blueprints for data. Information Systems Management. Winter. Pp. 90-92.

van Everdingen Y, van Hillegersberg J, Waarts E (2000) Enterprise Resource Planning: ERP Adoption by European Midsize Companies. Commun ACM 43 (4):27-31.

Vernadat Francois B. (1999). " Enterprise Modeling and Integration (EMI): Current Status and Research Perspectives". Annual Review. Vol. 26. Pp 15 – 25.

Venkatraman, N. (1991). IT induced business re-configuration. In Scott Morton M S. ed. *The Corporation of the 1990s: Information Technology and Organizational Transformation*. New York, Oxford Universities Press. Pp. 122-158.

Wand, Y & Weber, R. (2002). Research Commentary: Information systems and conceptual modeling – a research agenda. Information Systems Research. Vol. 13, No 4. Pp. 363-378.

Ward, J & Peppard, J (2002). Strategic Planning for Information Systems. 3rd Edition. Uk. John Wiley and Sons 624p.

Ward, J M. (1990). A portfolio approach to evaluating information systems investments and setting priorities. *Journal of Information Technology*. Vol. 5, No. 4. Pp. 222-231.

Weill, P, Broadbent, M & St cLair, D.R. (1996). IT value and the role of IT infrastructure investments. In Luftman, J.N (edit) Competing in the Information age. Strategic Alignment in Practice. Oxford, NY, Oxford University Press. Pp. 361-384.

Weill, P. & Broadbent, M. (1998). Leveraging The New Infrastructure: How Market Leaders Capitalize on Information Technology. Harvard Business School Press. Pp. 291.

Wißotzki, M., & Sonnenberger, A.(2012) Enterprise Architecture Management - State of Research Analysis & a Comparison of Selected Approaches. In PoEM, 2012 (pp. 37).

Whiteman,L.E, Huff,B.L & Presley,A.R (1998). The needs and issues associated with representing and integrating multiple views of the enterprise. In Information Infrastructure Systems for Manufacturing II. Chapman and Hall.

Williams, Theodore J & Litting. (1998) PERA AND GERAM- Enterprise Reference Architectures in Enterprise Integration. DIISM. Kluner Academic Publishers Institute for Interdisciplinary Engineering Studies, Purdue University.

Williams, T & Li, H. (1993). A Specification Statement of requirements for GERAM with all requirements illustrated by example from PERA. Report number 159, Purdue Laboratory for Applied Industrial Control.

Wiseman, C. (1985). Strategy and Computers. Homewood, Illinois, Dow Jones-Irwin.

Zach, O. (2012). ERP System Implementation in Small and Medium-Sized Enterprises. University of Agder, Agder.

Zachman, J A. (1987). A framework for information systems architecture. *IBM Systems journal*, Vol. 26, No. 3. Pp. 276-292.

Zachman, J.A. (1997). Enterprise Architecture: The Issue of the Century. Database Programming and Design. March 2-5.

Zuse H. (1990). *Software Complexity Measures and Models*, New York, NY: de Gruyter & Co.

APPENDIX I

SAMPLE OUTPUT

Customer Listing Report

S/N	Customer ID	Company	Contact	Address	City	State	Country	Telephone	Fax	stomer Type	ales Account	Discount	Credit Limit
1	C0001	ADDROCK NIG LTD	S00454	I 2 EDIDI LANE IDUMOTA	LAGOS	Lagos	Nigeria	08032214756		CASH	T2000	0.02	5,000
2	C0002	KONAS NIG LTD	EMMA	NO 2 POUNDS ROAD	ABA	Abia	Nigeria	08041252222		CASH	M2400	0.02	5,000
3	C0003	VILLOS ENTERPRISES	S0455	NO 13 POUND ROAD ABA	ABA	Abia	Nigeria	08032212566		CASH	M2400	0.02	200,000

PAYROLL REGISTER

Emp ID	Name	Ref	Date	Pay Type	Hours	Allowance (=N=)	Deduction (=N=)	Rate (=N=)	Tax (=N=)	Amount (=N=)
S0100	NWEKE GODWIN	RF001	04/03/2012	MONTHLY	0	80,000	2,500	65,000	5.5	142,494.5
S0102	BOROMA WILSON	RF002	04/03/2012	MONTHLY	0	56,000	1,800	55,000	4.2	109,195.8
S0455	OKEKE ANTHONY	RF003	04/03/2012	MONTHLY	0	12,000	1,000	15,000	.2	25,999.8
S00454	IKE JOSEPH	RF004	04/03/2012	MONTHLY	0	56,000	20,000	24,000	5	59,995
TOTAL						**204,000**	**25,300**	**25,300**	**14.9**	**337,685.1**

VENDOR TRANSACTION SUMMARY

S/N	Vendor ID	Name	Debit (=N=)	Credit (=N=)	Balance (=N=)
1	V0001	trans on INV: IVN0002	13,880.04	18,902	5,021.96
2	V0002	trans on INV: INV2001	4,002.5	200,000	195,997.5
3	V0003	trans on INV: INV2002	4,002.55	200,002.5	195,999.95
		Total	**21,885.09**	**418,904.5**	**397,019.41**

PAYROLL JOURNAL

Date	Employee	Account ID	Reference	Description	Debit (=N=)	Credit (=N=)
04/03/2012	S0100	M2400	RF001	Wages	142,494.5	
04/03/2012	S0100	M2400	RF001	Tax on Wages	5.5	
04/03/2012	S0100	T2000	RF001	Employee Pay		142,494.5
04/03/2012	S0100	T2000	RF001	Employee Pay		5.5
04/03/2012	S0102	M2400	RF002	Wages	109,195.8	
04/03/2012	S0102	M2400	RF002	Tax on Wages	4.2	
04/03/2012	S0102	T2000	RF002	Employee Pay		109,195.8
04/03/2012	S0102	T2000	RF002	Employee Pay		4.2
04/03/2012	S0455	M2400	RF003	Wages	25,999.8	
04/03/2012	S0455	M2400	RF003	Tax on Wages	.2	
04/03/2012	S0455	T2000	RF003	Employee Pay		25,999.8
04/03/2012	S0455	T2000	RF003	Employee Pay		.2
04/03/2012	S00454	M2400	RF004	Wages	59,995	
04/03/2012	S00454	M2400	RF004	Tax on Wages	5	
04/03/2012	S00454	T2000	RF004	Employee Pay		59,995
04/03/2012	S00454	T2000	RF004	Employee Pay		5
					337,700	**337,700**

ALL SALES ACCOUNT

S/N	Date	Description	Ref	Debit (=N=)	Credit (=N=)	Balance (=N=)
B/F						-25,999.6
1	04/03/2012	INV: C0001	1		20,000	-45,999.6
2	04/03/2012	INV: C0001	1	20,002	2,500	-28,497.6
3	04/03/2012	INV: C0001	1		400.04	-28,897.64
4	04/03/2012	INV: C0001	1	2		-28,895.64
5	04/03/2012	INV: C0003	2		7,200	-36,095.64
6	04/03/2012	INV: C0003	2		32,400	-68,495.64
7	04/03/2012	INV: C0003	2		40,000	-108,495.64
8	04/03/2012	INV: C0003	2	79,602	70,000	-98,893.64
9	04/03/2012	INV: C0003	2		1,592.04	-100,485.68
10	04/03/2012	INV: C0003	2	2		-100,483.68
11	04/03/2012	PAYROLL: S0100	RF001	5.5	142,494.5	-242,972.68
12	04/03/2012	PAYROLL: S0102	RF002	4.2	109,195.8	-352,164.28
13	04/03/2012	PAYROLL: S0455	RF003	.2	25,999.8	-378,163.88
14	04/03/2012	PAYROLL: S00454	RF004	5	59,995	-438,153.88
TOTAL				**73,623.3**	**511,777.18**	**-438,153.88**

CASH SALES ACCOUNT

S/N	Date	Description	Ref	Debit (=N=)	Credit (=N=)	Balance (=N=)
B/F						-20,976.04
1	04/03/2012	INV: V0002		196,000		175,023.96
2	04/03/2012	INV: V0003		196,002.45		371,026.41
3	04/03/2012	SALE INV: C0001	1	2,500		373,526.41
4	04/03/2012	SALE INV: C0003	2	70,000		443,526.41
5	04/03/2012	PAYROLL: S0100	RF001		142,494.5	301,031.91
6	04/03/2012	PAYROLL: S0100	RF001		5.5	301,026.41
7	04/03/2012	PAYROLL: S0102	RF002		109,195.8	191,830.61
8	04/03/2012	PAYROLL: S0102	RF002		4.2	191,826.41
9	04/03/2012	PAYROLL: S0455	RF003		25,999.8	165,826.61
10	04/03/2012	PAYROLL: S0455	RF003		.2	165,826.41
11	04/03/2012	PAYROLL: S00454	RF004		59,995	105,831.41
12	04/03/2012	PAYROLL: S00454	RF004		5	105,826.41
TOTAL				**443,526.41**	**337,700**	**105,826.41**

Trial Balance

TRIAL BALANCE : 04/03/2012

Account ID	*Description*	*Debit*	*Credit*
T2000	CASH FOR SALES	-469,526.41	
T2001	SALES ACCOUNT	-72,432.08	
M2400	SALES ACCOUNT		74,484.08
P1000	PURCHASE ACCOUNT		397,023.41

Cost Analyses. Stock Items TILES

INV.	*Item ID*	*Description*	*Date*	*Qty Sold*	*U. Cost*	*Total Cost*	*U. Price*	*Amount*	*Bal*	*Stock Bal.*
2	P1010	BLOCK TILES	4/3/2012	12	270	3,240	2700	32,400	29,160	58
2	P1000	TRIANGULAR TILES	4/3/2012	2	3600	7,200	3600	7,200		48
1	P2000	CIRCULAR TILES	4/3/2012	5	4000	20,000	4000	20,000		45
2	P2000	CIRCULAR TILES	4/3/2012	10	4000	40,000	4000	40,000		35
Total						70,440		99,600	29,160	

STOCK ITEMS PURCHASED BETWEEN: 04/03/2012 - 04/03/2012

S/N	*Date*	*Vendor*	*Inv. Number*	*Description*	*Qty*	*Unit Cost*	*Total Cost*
1	4/3/2012	V0002	INV2001	CIRCULAR TILES	50	4,000	200,000
2	4/3/2012	V0003	INV2002	TRIANGULAR TILES	50	3,600	200,000
				Total			400,000

APPENDIX II

SOURCE CODE

```
    Dim queryResult As rdoResultset
    Dim i As Long
    Dim totalBalance As Double
    Dim totalCredits As Double
    Dim totalDebits As Double

    rdo.rdoEngine.rdoErrors.Clear

    totalBalance = 0
    totalCredits = 0
    totalDebits = 0

    i = 1
    queryString = "SELECT * FROM accdetail WHERE vendorid LIKE ? AND accountid LIKE ?
AND transactiondate < ? ORDER BY transactiondate ASC"
    Set queryObject = rdoConn.CreateQuery("", queryString)

    queryObject.rdoParameters(0) = customerID
    queryObject.rdoParameters(1) = accountID
    queryObject.rdoParameters(2) = CDate(fromDate)

    Set queryResult = queryObject.OpenResultset(rdOpenKeyset)

    queryResult.MoveFirst
    Do While Not queryResult.EOF
        totalDebits = totalDebits + queryResult.rdoColumns(2).value
        totalCredits = totalCredits + queryResult.rdoColumns(3).value

        queryResult.MoveNext
    Loop

    getCustomerPreviousBalance = totalDebits - totalCredits
    queryResult.Close
    queryObject.Close

    Exit Function

errhandler:
     If rdo.rdoEngine.rdoErrors.Count > 0 Then
        For Each rError In rdoErrors
          'MsgBox "Error Number: " & rError.Number & "     " & rError.description & " Source: " &
rError.Source
        Next
```

```
 End If
 getCustomerPreviousBalance = 0

End Function
Public Function getPreviousAccountBalance(accountID As String, currentDate As Date) As Double
  On Error GoTo errhandler

  Dim queryObject As rdoQuery
  Dim queryString As String
  Dim queryResult As rdoResultset

  Dim totalBalance As Double
  Dim totalCredits As Double
  Dim totalDebits As Double

  rdo.rdoEngine.rdoErrors.Clear

  totalBalance = 0
  totalCredits = 0
  totalDebits = 0

  queryString = "SELECT * FROM accdetail WHERE accountid LIKE ? AND transactiondate < ?
ORDER BY transactiondate ASC"
  Set queryObject = rdoConn.CreateQuery("", queryString)

  queryObject.rdoParameters(0) = accountID
  queryObject.rdoParameters(1) = CDate(currentDate)

  Set queryResult = queryObject.OpenResultset(rdOpenKeyset)

  If queryResult.EOF Then
    getPreviousAccountBalance = 0
    queryResult.Close
    queryObject.Close
  End If

  queryResult.MoveFirst
  Do While Not queryResult.EOF
    totalDebits = totalDebits + queryResult.rdoColumns(2).value
    totalCredits = totalCredits + queryResult.rdoColumns(3).value

    queryResult.MoveNext
  Loop

  queryResult.Close
  queryObject.Close
  getPreviousAccountBalance = totalDebits - totalCredits
```

```
"********************************************************************************
***************
"perform delete operations here
"********************************************************************************
****************

"delete the transactions of customers marked for deletion
Public Sub deleteCustomerTransactions()
    On Error GoTo errhandler

    Dim queryObject As rdoQuery
    Dim queryString As String
    Dim queryResult As rdoResultset
    Dim icount As Long

    Dim queryObject1 As rdoQuery
    Dim queryString1 As String
    Dim customerID As String

    createConnection
    queryString = "SELECT customerid FROM customers WHERE  status LIKE 'ERASED'"
    Set queryObject = rdoConn.CreateQuery("", queryString)
    Set queryResult = queryObject.OpenResultset(rdOpenKeyset)

    If Not queryResult.EOF Then
        queryResult.MoveFirst
        Do While Not queryResult.EOF
            customerID = queryResult.rdoColumns(0).value

            On Error Resume Next

            queryString1 = "DELETE FROM sales WHERE customerid LIKE ?"
            Set queryObject1 = rdoConn.CreateQuery("", queryString1)
            queryObject1.rdoParameters(0) = customerID
            queryObject1.Execute

            queryString1 = "DELETE FROM receipt WHERE ownerid LIKE ? AND receipttype LIKE
'FORCUSTOMER'"
            Set queryObject1 = rdoConn.CreateQuery("", queryString1)
            queryObject1.rdoParameters(0) = customerID
            queryObject1.Execute

            queryString1 = "DELETE FROM accdetail WHERE vendorid LIKE ?"
            Set queryObject1 = rdoConn.CreateQuery("", queryString1)
            queryObject1.rdoParameters(0) = customerID
            queryObject1.Execute
```

```
            queryResult.MoveNext
        Loop
    End If
    queryResult.Close
    queryObject.Close

    closeConnection
    Exit Sub

errhandler:
    closeConnection
    MsgBox "Error on customer deletion"
End Sub
```

Appendix III

Archimate Notations Used in the Metamodels

Name	Meaning	Notation
Association	Association models a relationship between objects that is not covered by another, more specific relationship	
Access	The access relationship models the access of behavioral concepts to business or data objects	
Used by	The Used by relationship models the use of service by process, functions or interactions and the access to interfaces by roles, components or collaborations	
Realization	The realization relationship links a logical entity with a more concrete entity that it realizes.	
Assignment	The assignment relationship links units of behaviour with active elements (eg. Roles, components) that perform them or roles with actors that fulfill them	
Aggregation	The aggregation relationship indicates that an object groups a number of other objects	
Composition	The composition relationship indicates that an object is comprised of one or more other objects	
Flow	The flow relationship describes the exchange or transfer or for example information or value between processes, functions, interactions and events	
Triggering	The triggering relationship describes the temporal or casual relationship between processes, functions, interactions and events.	
Grouping	The grouping relationship indicates that objects of the same type or different types belong together based on some common characteristics	
Junction	A junction is used to connect relationships of the same type	
Specialization	The specialization relationship indicates that an object is a specialization of another object.	
Govern		
Standardize		

Mapping TIPS Module to Application Classification Model