SUCCESSFUL MODELLING OF THE ENTERPRISE

A guide for gaining insight into the enterprise

Christopher Magee

in close cooperation with:
Martin van den Berg
Marlies van Steenbergen
Daan Rijsenbrij

A cooperation between the Radboud University and Sogeti

VERSION INFORMATION

Version	Date	Remarks	Author
Draft_Version	March 2005	90% version	Magee, Christopher
Final_Version	May 2005	100% version	Magee, Christopher

DISTRIBUTION

Name	Organization
Daan Rijsenbrij	RU
Patrick van Bommel	RU
Marlies van Steenbergen	Sogeti
Martin van den Berg	Sogeti

PREFACE

To Whom It May Concern:

This report is a result of research conducted for the Radboud University of Nijmegen and Sogeti Nederland B.V. Our main research question was to determine how an architect can create a usable description of an enterprise. This research question was defined because architects require insight into the enterprise which enables them to develop a usable architecture. At the time this research initiated it was yet unknown how usable descriptions could be created.

The research has been conducted by ing. Christopher Magee as completion of the master course Information Science with specialization Information Architecture at the Radboud University Nijmegen. The research has been supervised by Prof. dr. Daan Rijsenbrij, dr. Patrick van Bommel, drs. Ir. Marlies van Steenbergen and drs. Martin van den Berg.

Firstly I would like to thank Sogeti Nederland B.V., especially Marlies van Steenbergen and Martin van den Berg, for providing me with the opportunity to conduct this research in an interesting environment, which Sogeti Nederland B.V has provided, and for the guidance I have been given during and after the research process. Secondly I would like to thank Daan Rijsenbrij for his guidance and support during the research process but most of all for his constant and enthusiastic challenging of the concepts developed by me for this research and on other subjects as it has contributed to my academic and personal development.

This research has been made possible by the contribution of a number of respected individuals of the digital architecture community. We would like to thank the following people as their time and enthusiasm have been of great influence on our research:

- *Marc de Goeij*, founder and account manager of Adaptive
- *Marc Lankhorst*, Telematica Institute and project leader of the Archimate project
- *Jaap Schekkerman*, thought leader for architecture at CapGemini and president of the Institute for Enterprise Architecture Developments
- *Hans Goedvolk*, consultant for CapGemini
- *Frits Broekema*, consultant for CapGemini
- *Monique Claus*, KPN
- *Rogier Dijk*, information manager for the Rabobank
- *Stijn Hoppenbrouwers*, Researcher for the Radboud University Nijmegen
- *Erik Proper*, Researcher and professor for the Radboud University Nijmegen
- *Patrick van Bommel,* Researcher for the Radboud University Nijmegen
- *Jan Hoogervorst*, consultant for Sogeti

- *Ivo van Ouwerkerk*, consultant for Sogeti
- *Leo Aarssen*, consultant for Sogeti
- *Stefan Langerveld*, consultant for Sogeti

Furthermore we would like to thank the following organizations as they have made a contribution to our research: BizzDesign, LogicaCMG, Ordina, Popkin Software and the University of Utrecht.

If you are interested in other research projects about Digital Architecture we advise you to view the following address: http://www.digital-architecture.net/scripties.htm

Best regards,

Christopher Anthony Magee
Amsterdam, May 2005

INDEX

ABSTRACT

Our research has been performed to determine how an architect can create a useable description of an enterprise. The research has been conducted on behalf of the Radboud University Nijmegen and Sogeti Nederland B.V. The research area is that of digital architecture, especially that of business architecture and information architecture.

Architecture demands a continuous process which consists, according to Sogeti, of three separate sub processes: that are as follows: strategic dialogue, architectural services and developing under architecture.

1. The strategic dialogue:
The strategic dialogue is a participation between general management and IT-management to determine the business objectives. These business objectives are translated into business cases with concrete project proposals. Each business case functions as a trigger for the next architectural process: architectural services.

2. Architectural services
Architectural services is a facilitating process that defines the enterprise architecture and architectures specific for projects. When architectural services is provided with a business case by the strategic dialogue it gathers relevant architectural principles from the enterprise architecture and translates them into project specific principles for a project. These project principles are defined in a project start architecture which is a guide that acts as boundaries for a project. If a project requires architectural principles that do not yet exist they are defined by architectural services. These newly defined architectural principles are then not only used by one project but they also become part of the enterprise architectural principles. By doing so the architecture becomes business case driven and project driven thus resulting in a Dynamic Enterprise because the enterprise architectural principles are constantly adapting to business demands and project capabilities.

3. Developing with architecture
In this process the business objectives, defined in the strategic dialogue, are realized. The projects are provided with a project start architecture in which the enterprise architectural principles are translated into specific architectural principles for a project.

To support the three processes we have researched the possibility of constructing models of the business architecture and information architecture. We have determined that modelling can contribute to gaining transparency of the organization which we believe is one of the elements that an adaptive enterprise requires.

Creating descriptions of an enterprise is an activity referred to as enterprise modelling. Three distinguishable types of enterprise modelling exist.

- Sketches are enterprise models that enable architects to help define a business case. Sketches are of strategic nature because they help describe the business objectives defined by general and IT-management in the strategic dialogue. Sketches can be used as a starting point for discussion or to gain architectural awareness throughout organizations.
- The second type of enterprise modelling is that of architectural modelling which offers the architectural design. Architectural models are descriptions of the enterprise which are best compared to building designs.

- The third and final type of enterprise modelling is business process modelling. Business process models offer descriptions of the business processes that exist with in an organization. Business processes form the operations of an organization and business process models help implement the architecture as it was designed in the architectural models.

Enterprise modelling, or at least architectural modelling and business process modelling, requires an enterprise modelling language which are predefined techniques that enable architects to construct enterprise models. Modelling languages are needed because they act as a guide for architects constructing the enterprise models and offer consistency to the users of the models. We have researched four of these modelling languages: UML, IDEF, ARIS and Archimate.
UML and IDEF are both modelling languages that originate from the area of system development which are now also used for business process modelling throughout the industry. ARIS is a modelling language that focuses on business process modelling and we found it to be the most usable one for this purpose. Archimate is a modelling language that can be used to create architectural models. We have determined that Archimate is a usable modelling language and that it is currently also the only architectural modelling language. Furthermore tools can and should be used to create enterprise models. It is however important to handle with care when making a choice for a tool as they can differ widely.

Along with the types of enterprise modelling we have determined that three types of architectural roles exist:

1. In a strategic role an architect can make use of sketches during the strategic dialogue.
2. In a tactical role and architect can make use of architectural models as part of the architectural services
3. In an operational role an architect can make use of business process modelling when developing under architecture.

To help the architect determine which type of enterprise model should be constructed we have introduced six guidelines that are based on level of use, architectural project group, user group, architectural awareness, quantity of details required and the architectural process.

1 CHAPTER ONE: INTRODUCTION

This chapter has as objective to inform the reader about the conducted research on how to model an enterprise. We will explain the content, by whom and why the research has been conducted. Secondly we will discuss the scientific areas in which the research has been conducted as it forms a basis for this report. We will conclude by describing the content and structure of this report.

1.1 General research information

1.1.1 Target group

This report is intended for architects of midsized to large organizations, architects that act as consultants to midsized and large organizations and for students, professors and other researchers with an interest in computer science, information science or business administration.

1.1.2 Constituent

The research has been conducted for the Architecture & Business Solutions division of Sogeti Nederland B.V, especially for drs. ir. Marlies van Steenbergen and drs. Martin van den Berg, and for the Information Retrieval and Information Systems research group of the Radboud University of Nijmegen, especially for Prof. Dr. Daan Rijsenbrij.

1.1.3 Researchers and Authors

The research has been conducted by Christopher Magee. Christopher has a Bachelors degree in computer science which he obtained at the Fontys University of Eindhoven. At this moment he is completing a Masters course in Information Architecture at the Radboud University of Nijmegen.

1.2 Research Area

1.2.1 Dynamic architecture

Before explaining the research in detail we will first discuss the research area as it is a required basis for understanding the research objectives. The research has been conducted in the field of digital architecture. Architecture for the digital world is a relatively young field of science which has not yet been clearly defined. At this moment in time there are numerous definitions and visions on digital architecture. This research has been conducted for and with Sogeti Nederland B.V. and we will use their architectural vision, as described in [DYA05] and [DYA04], as basis for the research as it is an open standard that offers a description of the architectural process which is required for our research. If we look at [Rijsenbrij04] we can determine that it offers the authors vision on what architecture is and also how it should be approached. The authors have described several approaches like e.g. the spiral approach but the actual execution is not discussed. The DYA method is an open

standard because not only the author's vision but also the execution of the architectural process is discussed. As our research has been conducted to support the process [Rijsenbrij04] is of little use to us as we know nothing about its execution. The Dynamic Enterprise Architecture method, or DYA, was developed by Sogeti Nederland B.V. and is a method for working under architecture. DYA has the following definition for architecture which will be the leading one throughout this thesis:

'A consistent set of rules and models that guides the design and implementation of processes, organizational structures, information flows, and technical infrastructure within an organization.'

This definition indicates that architecture is an instrument for general management and IT-management as it acts as a guide for making decisions that involve design and implementation. Architecture also acts as a guide for the people working under architecture as it creates boundaries for projects. These boundaries ensure that the projects deliver results that are part of the overall business strategy. Architecture is described by making use of rules and models. As we see architecture does not concentrate on a single part but on all aspects of an organization.

1.2.2 Architectural domains

According to the DYA method architecture is divided into separate architectural domains. In DYA there are three distinct architectural domains, they are as follows:

Business architecture
The architecture that sketches the organizational contours necessary to achieve the business objectives of the organization. Business architecture consists of three (architectural) objects (described in section 1.2.3).

Information architecture
The architecture that sketches the informational contours necessary to provide the organization with the information it requires. Information architecture consist of two (architectural) objects (described in section 1.2.3).

Technical architecture
The architecture that sketches the contours for implementing the technical infrastructure required by the organization. Technical architecture consists of three (architectural) objects (described in section 1.2.3).

In [DYA04] the authors have not described a solid basis for separating the architecture into three domain architectures but they define this as a common way of distinguishing the domain architectures. [Schekkerman] mentioned two domains, that of business and technology. We feel that the information domain is an essential attribute of architecture as it is not part of the business but a result thereof. In [Rijsenbrij04] the authors discuss four architectural aspects which are business, information, information systems and infrastructure. We feel that this is incorrect because information is actually data structured by the use of information systems. Because data is structured by the use of information systems, or actually applications as not all information systems are applications, data and applications together should form the information architecture. We can conclude that the separation into three architectural domains as defined by DYA is valid even though a reason for the separation is not offered.

1.2.3 Architectural objects

Each architectural domain consists of one or more architectural objects which are distinguishable entities within a domain architecture. Each object has clear boundaries set to distinguish a particular concept. The first architectural domain, business architecture, has the following architectural objects:

Product/ service
The product and or service architecture describes the products and or services that an organization offers. This concerns attributes like brand, the structure of a product, time-to-market of the product etc.

Process
The process architecture describes the processes and their relations as they exist within an organization. The process architecture mainly focuses on the more high-level processes of an organization.

Organization
The organization architecture prescribes how the organization and the employees are organized, together with their relations to the external partners and service providers. In [Schekkerman04] we found that most definitions on the organization architecture are broader. In [Rijsenbrij04] architecture consists of four architectural aspects of which one is business. The following definition is provided:

‘ The business aspect is a total description of the mission, vision and objectives of an organization, the structure and relations of an organization, the organization and value chain, organizational rules, tasks and activities’

The definition provided by [Rijsenbrij04] mentions the mission and vision of an organization which are not part of DYA. Because we feel that the organizational mission and vision are a crucial part of architecture, organizational success depends on a correct mission and vision, we will from this point forward use the following definition which is the original DYA definition extended with the mission and vision:

The organization architecture describes the mission, vision and how the organization and the employees are organized, together with their relations to the external partners and service providers’

Compared to the DYA definition for an organization we see that this definition also encompasses tasks and activities. In DYA the tasks and activities are part of the process architecture and we see no reason to alter this as it is a matter of choice. The second architectural domain, information architecture, consists of the following architectural objects:

Data
This architectural object describes the storage, maintenance and use of the for the organization relevant data. The data architecture describes which data is relevant for the organization together wit the structure and internal relations and how the responsibility and organization of the data is organized.

Application
This architectural object describes the use of information systems within an organization. The application architecture describes the different information systems and their relations within the organization together with the principles for use, design and implementations of applications.

The third and final architectural domain, technical infrastructure, consists of the following architectural objects:

Middleware
This architectural object describes the software components and their relations, within an organization, that make it possible for applications and users to communicate with each other.

Platform
This architectural object describes the hardware of an organization. The hardware consists of mainframes, desktops, terminals and other equipment including their operating systems.

Network
This architectural object describes the principles and models that are of influence to the connectivity of the network(LAN/WAN).

In the DYA definition for architecture we read that the architectural objects organization and processes are mentioned. A remark we can make is that the architectural objects product/service and applications are both *not* mentioned in the DYA definition for architecture. Furthermore the data and application architectural objects are suddenly replaced by information flows. We have also determined that the architectural objects middleware, platform and network have been generalized to technical infrastructure. We advise the authors of [DYA04] to encompasses all the architectural objects into the definition.

1.2.4 Levels of architectural abstractions

Besides the three domains of architecture DYA also divides architecture into the following three levels of abstraction:

General principles
Principles which combined form a reflection of the vision of business and IT management. Providing the customer with a single point of contact for his or her enquiries is an example of a general principle.

Rules and guidelines
General principles are translated into concrete rules and guidelines for each domain architecture. An example of a rule is to make use of a standard such as XML. A remark that we must make is that the original version of DYA [DYA01] mentions only rules. The English translation in [DYA04] has been extended with guidelines but they are not yet mentioned in the DYA definition.

Models
Diagrams and descriptions of current and future (desirable) situations.

1.2.5 Architectural framework

Frameworks are logical structures for classifying and organizing information in such a way that a basis for further analysing is created. In [DYA04] a framework, called the DYA architectural framework, is introduced in which the architectural domains and the architectural levels of abstraction have been structured. This architectural framework or actually the matrix in which it has resulted offers a basis for further analysing and during this report we will make extensive use of it. The matrix is divided into the three domain architectures, which are subdivided into object architectures, and into the three conceptual levels. The domain architectures make up the columns of the matrix, while the conceptual levels constitute the rows of the matrix

1.2.6 Research domain

Our research has focused on a specific region of the architectural framework. The exact domain is two of the three architectural domains, the business architecture and the information architecture, and the third architectural abstraction, models, of the DYA architectural method. In figure 1.1 we have positioned the research domain on the DYA architectural framework.

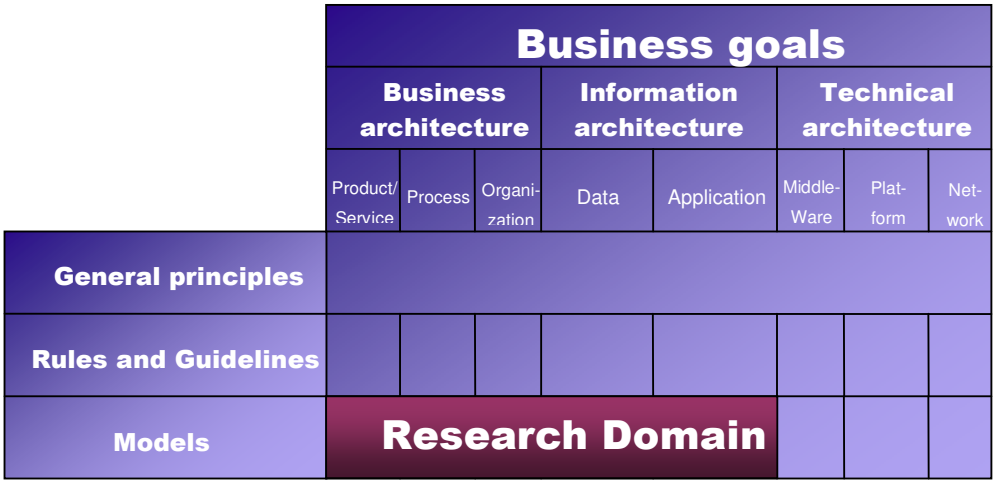

Figure 1.1: DYA Architectural framework and the research domain

1.2.7 Scope

The research has focused only on the domains of business architecture and information architecture, the third domain technical infrastructure architecture has not been researched. This choice has been made due to time related issues. Furthermore the research was initially defined to research only modelling languages and not modelling tools as such research has already been conducted by the constituent [Burgt04]. It is however the case that during the research we found that modelling languages and tools are in some occasions inseparable thus we will also discus several tools in this paper.

1.3 Research information

1.3.1 Cause & Problem Definition

This research has been conducted because a lack of insight exists into the quantity and quality of modelling languages that can be used for modelling the enterprise. Architects are faced with different modelling languages of which they do not know the objectives or how they can be used. The following problem has been defined:

'Within the world of digital architecture there is a lack of insight into modelling the enterprise and on the several modelling languages that can be used to model an enterprise'

1.3.2 Relevance

There are several points that indicate the relevance of this research. The first is that the need for this research has originated from Sogeti Nederland B.V where architects practice as consultants to other organizations. When performing an architectural study the architects often have the need to create models that can assist them during the architectural study. Before this research was conducted they felt that they weren't able to make a choice for a modelling language based upon solid knowledge.

The second reason is that architects that function within an organization, not at as consultants to an organization, have the need for modelling languages that can offer support during an architectural project. Further they have a need to communicate the architecture of their organization.

1.3.3 Research questions

The main research questions is as follows:

'How can an architect create a usable description of an architecture?'

The main question has resulted in the following sub questions:

- Which modelling languages are currently available?
- What are the properties of these modelling languages?
- What are the main objectives of the modelling languages?
- What are the main quality factors of modelling languages?
- Is there any experience with the modelling languages?
- What are the influences that affect the choice for a modelling language?
- Is it possible to combine enterprise modelling languages?
- What are the consequences of making use of an enterprise modelling language?

Our research has provided the answers for the research questions which will be discussed in this report.

1.3.4 Objectives

This research has resulted in the information that is required by the architects. The research objectives are as follows:

Set of quality criteria
The criteria will form a basis of judgment of the modelling languages. The criteria will be used to judge the quality of the modelling languages.

Set of usable modelling languages
This research has resulted in a set of modelling languages that can be used for modelling of enterprise. The set of modelling languages has been judged by making use of the criteria.

Guidelines
The research has resulted in some guidelines which an architect can use when there is a need to model an organization. By making use of the guidelines an architect can determine which type of modelling language can be used.

1.3.5 Deliverables

Because the research has been undertaken for two parties the deliverables have been spilt. The research has three deliverables that are as follows:

Master thesis
The master thesis describes the reasons for this research, the research domain, the research results and the research process.

Sogeti rapport
The Sogeti report provides detailed information on the researched modelling languages and gives an advice for the use of a modelling language.

Article for publishing
After the master thesis and the Sogeti report have been finished and approved by the constituents an article will be written. This article will be a compact version of the research results and a publisher will be sought.

1.4 DYA model

1.4.1 Architectural process

DYA not only defines architecture but it also describes how architecture should be used. As our research, modelling, is part of the entire architectural process we will briefly discuss the entire architectural process according to the DYA method.

1.4.2 Speed & Alignment

Organizations face the constant pressure of the market which causes them to constantly transform their business. As business processes are highly dependent on information technology it is also necessary to optimize the use of the information

technology. A problem that arises, is that the pressure for change, demands fast optimizing of the business. This cannot always be aligned properly with changes in the (information) technology as these changes take more time. The information technology exists out of software and hardware that have been constructed to last for several years or even decades. Business demands on the other hand have been constantly changing. Over time the information technology has been updated to meet the business demands in such a way that large complex networks of systems have evolved. When a small update is required the entire network has to be analysed and updated which requires large amounts of time and resources. An example is that of [KPN04] that describes how over several years such a large network of systems has evolved at KPN. A result is that customer satisfaction is dropping because KPN could no longer satisfy their needs due to complex information technology at KPN. The DYA architectural method focuses on optimizing the balance between the time in which changes need to be made and the alignment between the business and IT. To encompass this alignment DYA foresees in three separate architectural processes: Strategic dialogue, architectural services and developing with(out) architecture. Together these three processes ensure that the architecture is driven by business demands and project capabilities as together they form the basis for defining the architectural principles thus an dynamic architecture is created. The DYA method is somewhat unique by doing so as other methods only describe the need for architecture but not how it should be performed. [Hoogervorst] also mentioned this as he says that published materials all refer to the need for architectural principles but they hardly discuss how these principles should be defined. According to [Dijk] this has also been his main driver for the Rabobank to make use of the DYA method. In the following sections these architectural processes are discussed.

1.4.3 Strategic Dialogue

In the strategic dialogue the business objectives are defined and transformed into business cases with concrete project proposals. Both the business management and the IT-management participate in the strategic dialogue and together they determine the business objectives. The business case teams then create business cases which describe how to achieve the business objectives, what the impact might be on the organization and what the financial benefits will be. When a business case is approved, a concrete project proposal is created. In the strategic dialogue the participation of both business management and IT-management is crucial for success because success is dependent on the business needs matching with the IT capabilities. The business cases are provided to the next architectural process, architectural services, that develops the enterprise architecture and architectures specific for projects. By providing the business cases to the architectural services a business case driven architecture is created as it constantly adapts to the business needs.

1.4.4 Architectural services

Architectural services is the second process of the entire architectural process. The objective of this process is to define the architectures. Architectural services is a facilitating process to the first architectural process, strategic dialogue, and the third architectural process, developing with(out) architecture. A business case created in the strategic dialogue forms the trigger for the architectural services process. The defined architectures are made available to the participants in the strategic dialogue and to the projects.

Architectural services is a facilitating process that defines the enterprise architecture and architectures specific for projects. When architectural services is provided with a business case by the strategic dialogue it gathers relevant architectural principles from the enterprise architecture and translates them into project specific principles for a project. If a project requires architectural principles that do not yet exist they are defined by architectural services. These newly defined architectural principles are then not only used by one project but they also become part of the enterprise architectural principles. If enterprise architectural principles are no longer used by projects they can be dismissed. A result is that the architecture becomes business driven as it constantly adapts to the business demands. By creating architectures for specific projects one is also sure of the architecture being effective thus architecture also becomes project driven.

1.4.5 Developing with(out) architecture

The third and final architectural process is developing with(out) architecture. In this process the business objectives, as set out in the strategic dialogue, are realized within the set time, with an acceptable level of quality and within budget.

A project team that is developing under architecture is provided with a document called the project start architecture, also known as a PSA. In a PSA the overall architectural principles and models are translated into specific project architectural principles. The project principles act as boundaries which projects must uphold to ensure that deliverables contribute to the overall business strategy.

As projects work within the boundaries defined in the form of architectural principles their feedback to architectural services is important. When architectural principles form an obstacle for the projects this can be communicated to architectural services. Architectural services can then make the appropriate adjustments and this is why the projects are an essential part of the architectural process.

In the 'real world' we know that it is not always possible to obtain such a high degree of architectural abiding, therefore, in unusual or time-pressured circumstances, it is possible to start developing without an architecture. Even though the project is initiated without an architectural framework, control is not totally lost. When a project is being performed a parallel activity is initiated that eventually will bring the project back under architecture.

1.4.6 Process Cycle

The three main processes of architecture have a cyclic nature. All processes are continuous and influence each other. Furthermore there can be multiple cycles at any given time. As a project might be realizing objectives set in a strategic dialogue, another strategic dialogue might already be in progress trying to identify new business opportunities. Governance is required for the entire process cycle as it is in the ability to control and direct. The process cycle is visualized in the DYA architectural model in figure 1.2.

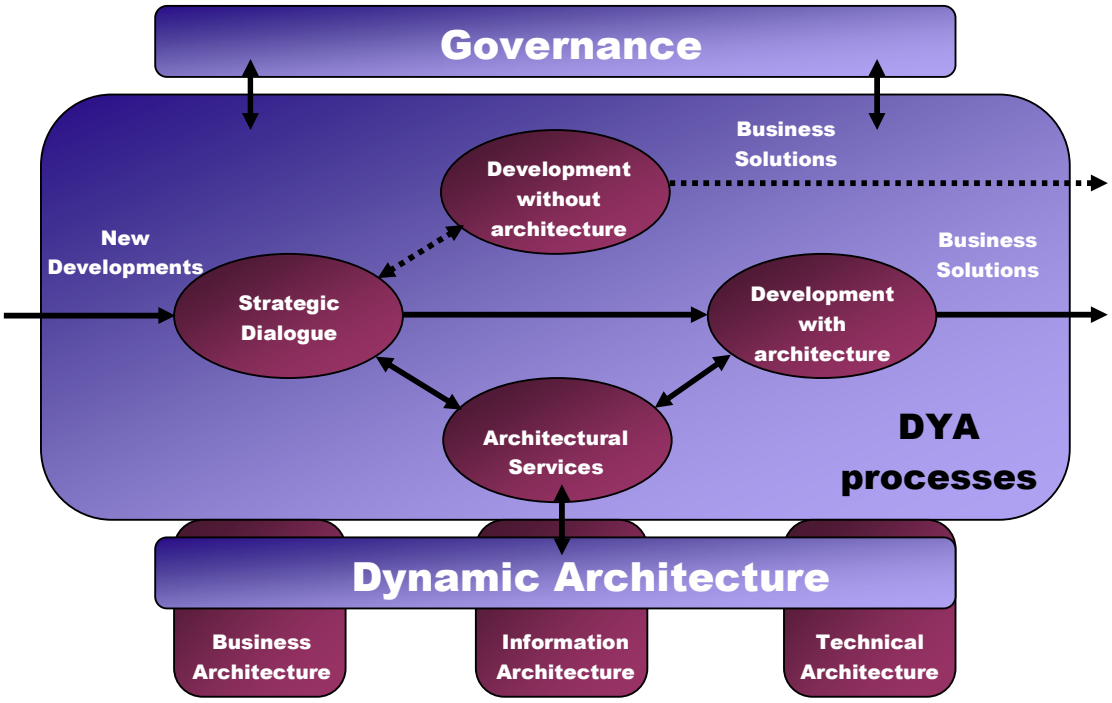

Figure 1.2: DYA architectural model

1.4.7 Just enough just in time principle

DYA has a process principle that should be upheld during the entire architectural process. This is the 'Just Enough Just in Time principle' which ensures that when performing the architectural process organizations do not treat the architectural framework as 'holy'. By this we mean that architects should not spend all their time focussing on the entire architectural framework but that they must identify which parts of the framework are crucial to succeed in realizing the business objectives.

1.5 Document structure

In the second chapter, modelling, we discus the basics of modelling, the current needs of organizations and how modelling can support organizations in obtaining their objectives. The term enterprise modelling is discussed together with the types of enterprise modelling. In the second chapter we will also introduce our Enterprise Modelling Framework which will play a leading role throughout this report.

The third chapter, modelling languages, describes several criteria that can be used to determine if a modelling language can be of use to an architect. Secondly we will position the modelling languages on our Enterprise Modelling Framework and discuss the similarities and differences between the several enterprise modelling languages. Finally we will apply our criteria to the enterprise modelling languages.

In chapter four, the practical aspects of modelling, we describe how modelling should be performed by introducing guidelines, what place modelling has as part of the entire

architectural process. Furthermore we will discuss the use of tools that can support modelling.

In the fifth and final chapter we will discuss our conclusions and recommendations that have been reached during the research.

If one is interested appendix A and appendix B describe the research process and a reflection on the research process.

2 CHAPTER TWO: MODELLING

2.1 Introduction

This chapter will inform the reader on the basics of modelling in general. Secondly we will discuss the current needs of organizations to see if they can benefit from the use of modelling. Finally we will describe a specific type of modelling known as enterprise modelling which is used to create descriptions of organizations.

2.2 Modelling

2.2.1 Definition

To understand modelling in general we will first discuss two definitions proposed by others and then define our own definition.

1. *'A model is a description of observed behaviour, simplified by ignoring certain details.' [Heritage]*

2. *'A model is a simplified mathematical description of a system, used to assist calculations and predictions.' [Oxford]*

Definition one implies that a model must be a description of behaviour without mentioning to who or what the behaviour belongs. We will assume that it is the behaviour of some sort of entity.

3. *'An entity is something that has a distinct, separate existence, though it need not be a material existence'. [Wikpedia]*

Definition three mentions that an entity is something existing but not necessarily in a material form. As we wish to create models of something that exists we have assumed that the behaviour belongs to an entity. Because an entity can also be of non material form it is possible to create models of something that e.g. exist only in our minds. As it will not always be possible to observe behaviour of non material entities we will disregard the behavioural concept as mentioned in definition one.

The second definition mentions a system. So what is a system? In [Ashby57] we read that a system is a collection of states with transformations between them and that the transformation from one state to another is called behaviour which is also mentioned in definition one. The behaviour displayed is depended on the input it receives from its external environment and internal state. We find the definition of a system a useful definition as it differentiates systems not by their name but by their behaviour. By doing so we can make comparisons between systems and determine if they are equal or that they differ from each other. We can also determine if a system behaves as we would aspect and it enables us to learn to understand the system. Further we can determine that systems are entities that exist within an environment as their displayed behaviour is triggered by their environment.

Furthermore definition two speaks of a mathematical description which can be true in some occasions but a description is not always of a mathematical nature. A painting or a photograph can e.g. be a description of a person's behaviour even though it is not a mathematical description. With this in mind we have defined a model as follows:

'A model is an abstract description of behaviour belonging to a system, simplified by ignoring certain detail.'

2.2.2 Goals

Knowing what models are we will discuss why models are used and what can be gained by their use. One could say that models are used to express aspects of some kind of reality, or a concept of reality.

According to [Wikpedia] models are used for the following objective:

'The purpose of a model is to provide an argumentative framework for applying logic and mathematics that can be independently evaluated (for example by testing) and that can be applied for reasoning in a range of situations'

An argumentative framework can be used to gain understanding of the unknown by placing the unknown in a framework so that the expressed can be understood.

In [Vermeulen, Aaldijk] we read that modelling has several objectives which lead to transparency, they are as follows:

1. Models offer a basis for analysis and argument
It is not possible to perform analysis or argue about something if there is no basis available. An example is architecture itself. Analysing architecture is of no use if there is no exchangeable definition of architecture. To gain this understanding several models of architecture have been proposed. An example of an architectural model is that of the DYA architectural framework which we discussed in the first chapter.

2. Models offer a basis for declaration and communication
Models provide a shareable representation of knowledge so communication is possible which is essential for achieving consensus. Consensus can be reached by users as they can make declarations and reach agreement on the models.

3. Models help change
If we construct models of the current and required state of a system the transformation process between the states can be defined as the models enable us to understand the current and required state.

4. Models reduce complexity
As models ignore certain detail they reduce the complexity which makes the described behaviour understandable. A lot of detail can be complex and it can be hard to understand a complex description.

2.3 Modelling an enterprise

2.3.1 Introduction

As modelling is creating a description of the behaviour belonging to a system, it is possible to create models of an enterprise as it also displays some form of behaviour. Before discussing this in more detail we will first determine the needs of an organization to see if it is useful to create models of an enterprise.

2.3.2 Organizational needs

Due to the present global and highly competitive market organizations seem to be changing at a high level of speed. Organizations are doing so to keep their, or gain new, business advantages over their competitors. Due to these changes organizations are constantly forced to align their business with the ever changing demands of the market. Organizations that are performing this process are referred to as the adaptive enterprise. In [HP03] we read:

'An Adaptive Enterprise is one that can quickly respond to and capitalize on change for business advantage. It is the ultimate state of fitness: business and IT perfectly synchronized'

According to the DYA method an adaptive enterprise, or dynamic architecture, is in according to the following description:

'Dynamic Architecture: An enterprise architecture that has been developed with speed of change in mind and, more specifically, the dynamics involved in facilitating change'

If we compare the two definitions we see that The DYA definition fails to describe that the Dynamic Architecture is required to gain business advantage which is of course the ultimate driver for business. The HP definition does mention that business advantage can be gained by synchronizing the business and IT. A remark we must make is that perfect synchronization will not automatically lead to business advantage as it depends on more factors.

Adaptive enterprises are now shifting their focus to their core processes as they are of the highest value to an organization. Support processes are being outsourced to specialized partners thus forming organizational networks as described in [Rijsenbrij04]. Due to the globalization these networks are of a complex nature. This complexity is often driven by the advance of IT [Gartner04]. In [Vernadat] we read that organizations are forced to gain insight into their own operations to deal with the constant transformations the enterprise undergoes. We have therefore defined the following rule for successful adaptiveness:

'Organizations that are transforming their operations can only be successful if they truly understand their own operations'

If an enterprise is willing to be adaptive it must understand its current state to determine if and which transformation is required. Our rule demonstrates that successful transformation depends on understanding the current state of the enterprise.

2.3.3 Transparency of the enterprise

In chapter one we described the example of [kpn04] which is a good example on the importance of transparency. KPN's customer satisfaction was dropping because KPN's information technology was no longer transparent. Within KPN a need for change had evolved but it was unclear what the change should be due to the lack of transparency. As we described modelling offers such transparency.

We have established that modelling is of use to an adaptive enterprise and we will now pay attention to enterprise modelling languages and the several types of enterprise modelling.

2.3.4 Enterprise modelling

An enterprise is a specialized form of a system as it also displays some kind of behaviour by transforming between several states. We will now replace the term system with the term enterprise in our definition. Further we have discussed the aspect of change several times. In our current modelling definition the transformation from the current state to the required state is not yet discussed. To involve the concept of change we will extend our definition with the current and required enterprise.

'An Enterprise model is an abstract description of the behaviour belonging to an existing or required enterprise, simplified by ignoring certain detail'

Enterprise modelling is done by making use of a modelling language. A modelling language consists of a collection of methods that together allow one to create models of an enterprise. The collection of methods has as objective the modelling of the entire enterprise. We conclude:

'An enterprise modelling language is a collection of methods with the objective of creating enterprise models'

2.3.5 *Types of enterprise modelling*

When we speak of enterprise modelling we speak of a generalized form of enterprise modelling. During the research we discovered that there are three distinctive types of enterprise modelling. The types of enterprise modelling are as follows:

- *Sketches*
  This type of enterprise modelling is used to describe the *'big picture'* (figure 2.1).

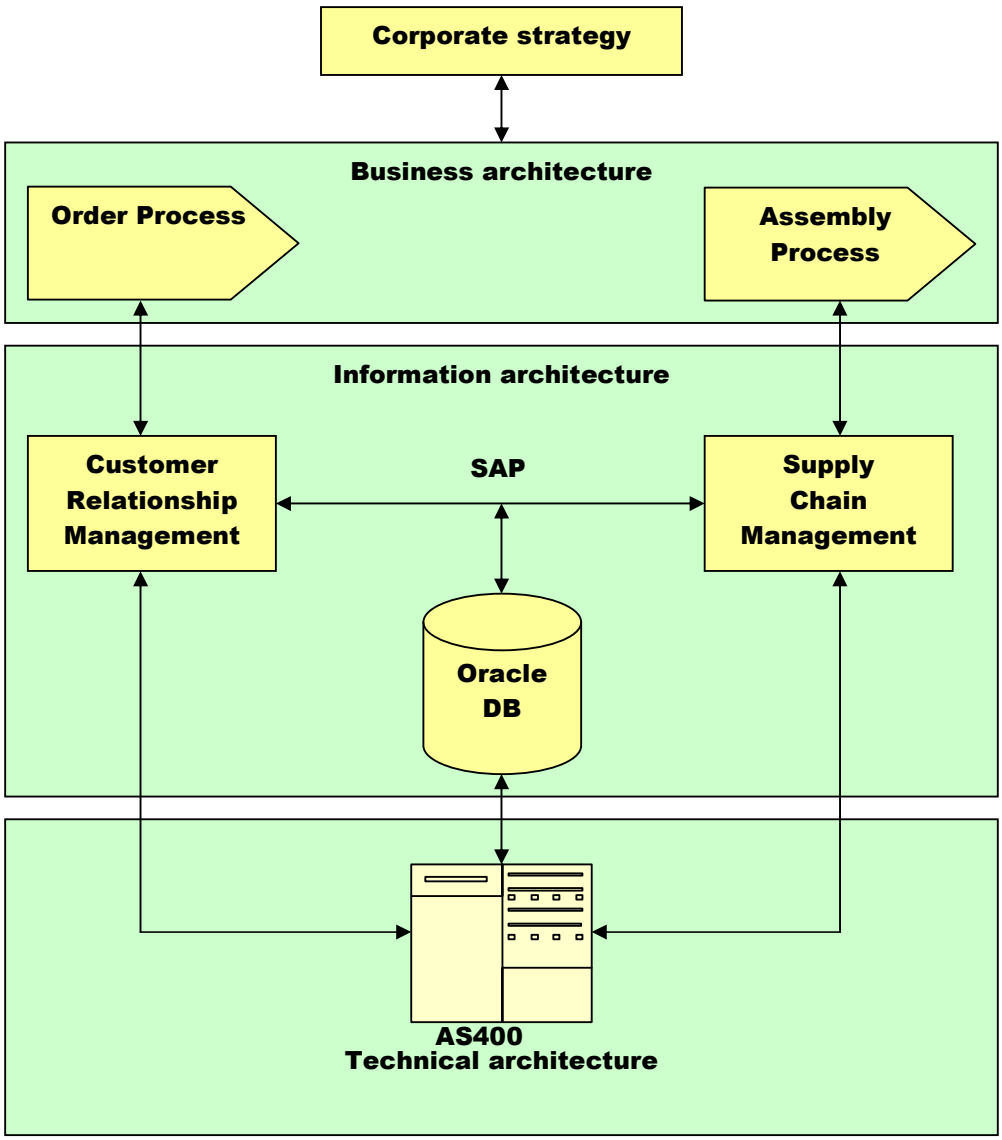

Figure 2.1: Sketch that describes the big picture of an organization

- *Architectural models*
  This type of enterprise modelling is used to determine *'what the architectural design looks like'*. Figure 2.2 is an architectural model designed for the ING bank [ING04].

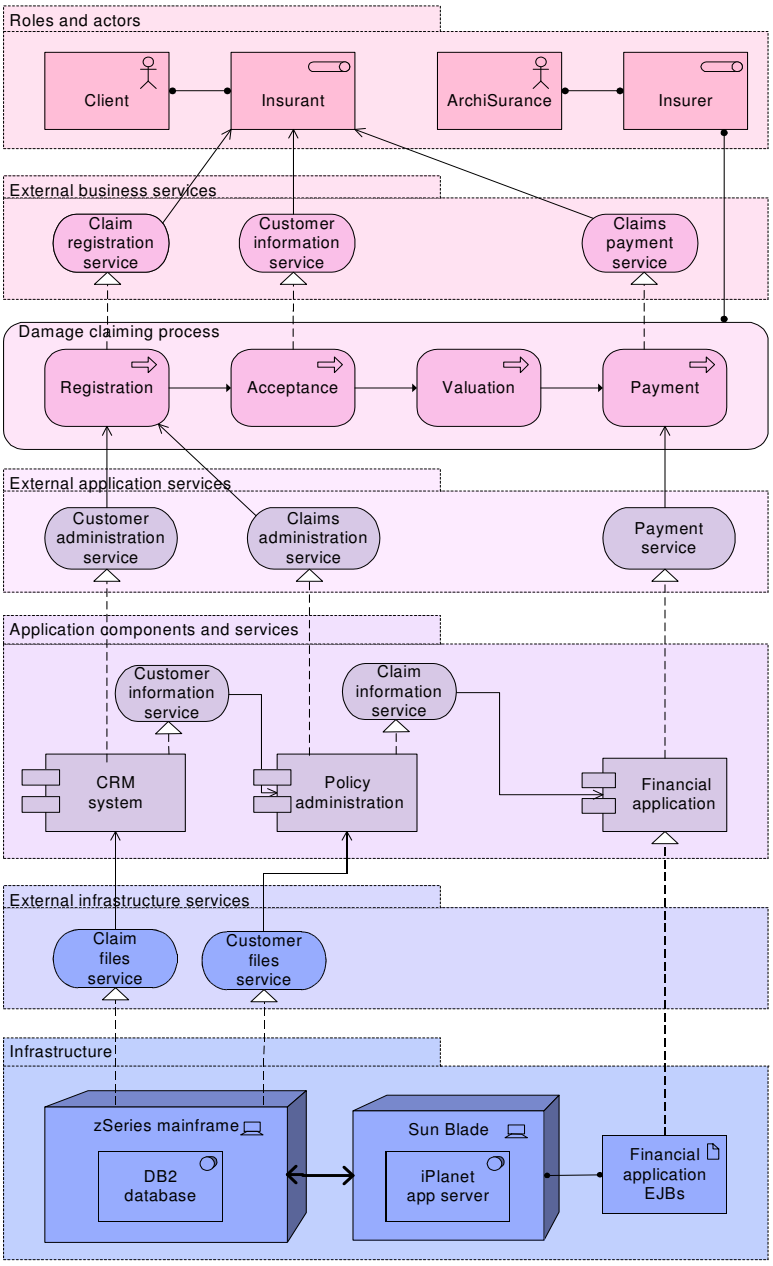

Figure 2.2: Architectural model

- *Business process models*
  This type of enterprise modelling is used to *'implement the architecture'*, demonstrated in figure 2.3 [Spiekerman03].

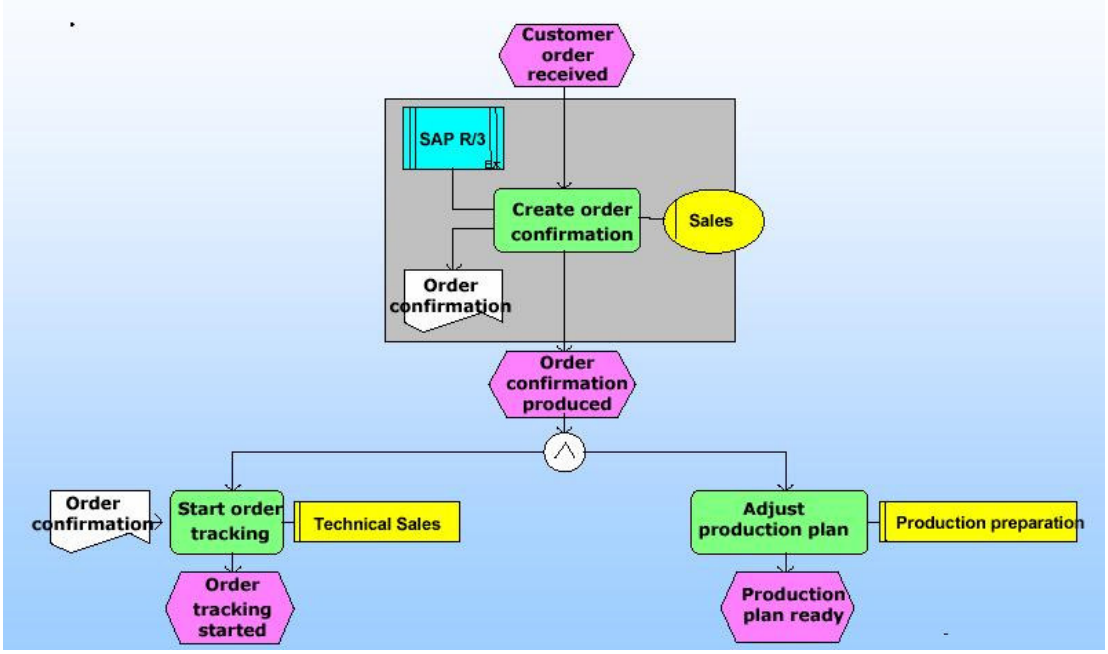

Figure 2.3: Business process model

Each type of enterprise modelling will be discussed in the following section.

2.4 Sketches

2.4.1 Introduction

Sketches are models that are produced in a small amount of time by disregarding a lot of detail. Sketches give brief and general descriptions that are often used as a preliminary study for the following reasons:

1. 'Sketches form a starting point for discussion'

Even though some architectural knowledge is available, it is usually scattered throughout the minds of individuals within organizations. Architects make use of sketches as a starting point for discussion as there is usually no material available yet.

2. 'Sketches are used to gain awareness of the need of architecture in an organization'

Architects are confronted with a lack of acceptance within the organizations. There are some defined architectural principles within the organization but architecture is not appreciated by higher executives throughout the organization. Architects make use of

sketches to demonstrate the use and benefits of architecture or the problems that exists throughout the organization due to a lack of architecture.

2.4.2 Paper drawings

In the interviews we discovered that almost all architects made use of paper sketches. These paper sketches are models in the form of simple drawings on a whiteboard or paper models that are created by making use of cardboard figures. The main reasons we found during our research for using these paper drawings are summarized below:

To gain insight into the architecture of an organization
Architects make use of these sketches to gain an insight into organizations architecture. As the architecture of an organization is complex, sketches have the ability to ignore lots of detail but still offer the main 'big picture' of an organization.

80 – 20 rule
By making use of sketches the architect can gain 80 percent knowledge of the organizations architecture in 20 percent of the time [Ouwerkerk]. This is correct as this 80 percent represents the big picture that was just discussed. The big picture is created by ignoring a large amount of details which makes it possible to construct them in a short amount of time.

Stakeholder perspective
Anyone can create a sketch as it is no more then a simple drawing. Making sketches also offers the stakeholder or other organizational employees the capability to demonstrate their own perspective of the architecture or part of the architecture

2.4.3 Digital drawings

The second form of sketches is of digital form. These models are often created by using a simple tool. The main objectives the architects want to achieve are summarized below:

Formal communication
The digital sketches are often used to communicate the architecture to a higher level of executives and are therefore constructed in a more formal way. Paper sketches look unprofessional and can make a bad impression and this is why digital sketches are constructed. Note that these models are formal in the sense that they are in accordance to the guidelines of contact that exists within an organizations boardroom, not in the sense that they abide by any form of mathematical logic or rules.

Persuasive
The digital sketches are more structured in comparison to the paper sketches. The importance of structured and impressive models is great as the message being told is just as important as the package it is delivered in.

Reusability
After use the models often need to be updated [Ouwerkerk]. Digital sketches provide us with models that have a higher degree of maintainability and are therefore more reusable. Reusable models are more time efficient.

2.4.4 Attributes of sketches

Generalists/ higher management
The sketches often target the generalists within an organization such as division managers or even higher executives.

Strategy & Objectives
As the sketches are used in the beginning of an architectural project or architectural process, they are useful in determining the business objectives that have to be achieved by making use of architecture.

'Free' of semantics
The models are very informal in the sense that they hardly need to abide by any rules or guidelines that exist in formal modelling languages. The lacks of semantics don't form a problem as the higher level managers are more generalists than specialists who are not used to working with complicated models.

Subjective
As the models are free of rules and guidelines they become subjective. Models constructed by different architects will be somewhat inconsistent as the models will be constructed different. This does not mean that the models themselves are inconsistent but that different architects will construct different models of the same situation. Quality of the models is also is dependant on the architect's capabilities.

2.4.5 Advantages of sketches

Time related
Paper sketches can be created at short notice which is especially important during the strategic dialogue. As business demands change rapidly the strategic dialogue must take a minimum amount of time to define business objectiveness so business demands can be met on time.

Creativity
As sketches have no rules or guidelines an advantage of the sketches is that they can be created however an architect deems necessary.

Customized
Architects can create models that are 'custom made' for an organization. This has a positive effect on communication.

2.4.6 Disadvantages of sketches

Transfer
A problem with sketches is that they are not easily transferable between architects. As the nature of sketches is subjective they can only be transferred to another architect if accompanied by oral and or written communication.

Distribution
Another disadvantage is that the paper sketches are hard to distribute especially in large organizations. Digital sketches are of course easier to distribute.

Revisions
Paper sketches have a major disadvantage that one can not make use of revisions. As paper sketches cannot be reused, simple adjustments to a model in the form of a revision require that the entire model has to be created again instead of altering a copy of the model.

2.5 Architectural models

2.5.1 Introduction

The term architectural modelling was chosen because this type of enterprise modelling focuses on one of the main objectives of architecture: Alignment between the several architectural domains. Sketches will only describe the existence of alignment but architectural models also describe what the alignment looks like. Architectural models are used to determine *'what the architecture looks like'.*

2.5.2 Objectives

As is the case with the sketches, there are several reasons why architects use this second type of enterprise modelling, they have been summarized below:

To work out the sketches
The sketches are very informal and therefore less adequate for middle management who requires more information about the specific area in which they work.

Standardization
The sketches are hard to transform between architects, due to their subjective nature, there is a demand for more transferable models [Ouwerkerk]. This demand is justified as architects can e.g. become relocated on another project. Standardization of models will lead to more transference between architects.

Consistent structure
Sketches can differentiate largely and a more consistent structure is required for detailed analysis [Hoppenbrouwers]. When models become more consistent it can lead to a higher number of people that are able to read and understand the models without oral and textual explanation which justifies the demand for consistency.
Standardization and consistency can be reached by introducing a set of predefined rules which architects must uphold.

A basis for further detailed design is needed
The gap between an idea in a sketch and direct implementation is too large [Broekema]. When e.g. developers are confronted with only sketches they will not be able to construct plans to actually realize the architecture. This transformation process, form idea to implementation, requires a more detailed design which is hard to achieve if there are only sketches available. A '*step in between is needed*'.

Risk analysis
Architects wish to gain more control over their architecture by performing risk analysis or impact analysis [Goeij]. Impact analysis enables the architect to analyze and identify the results of changes made in the architecture. Sketches do not offer such capabilities but an architectural modelling language makes this more feasible.

Detail
Sketches help define a strategy. As the lack of information in these sketches prevents a solid basis for judgment more detailed models are required to help define a tactic.

2.5.3 Attributes of architectural models

Integration between the architectural domains
Architecture modelling focuses not only on a part but on the entire architecture and more importantly on the alignment between architectural domains.

Tactical by nature
The architectural models are of a tactical nature. As the strategy has been defined through the sketches the architectural models are used to help define the tactic. Sketches are used to communicate ideas; architectural models are used to work out these ideas.

Middle segment of an organization.
The architectural models usually target the several stakeholders that are part of the middle segment of an organization. These stakeholders are the ones in an organization who define the organizational tactic such as information managers, financial mangers operational managers.

Offer some impact analysis
Architectural modelling concentrates on the architectural domains and also on the alignment between several domains. Changes on one architectural domain inflict changes on the other which can be identified by the use of impact analysis. Impact analysis is automated support offered by tooling to help architects determine the impact of a specific change.

2.5.4 Advantages of architectural modelling

Transferable
When a type of modelling language becomes an industrial standard, an organizational standard or it at least offers some consistency, the result is that the work becomes more transferable between different architects.

Maintainable
Architectural models are more maintainable than the sketches. Architectural models are usually in digital form and a tool can be used that has been specifically developed for a modelling language. This also makes these models easier to work with.

Distribution & Storage
Because the architectural models are in digital form they can be easily distributed as part of a report or by email. The storage of the digital models, contrary to the paper sketches, becomes easier.

Revisions & Reuse
The digital form and tooling makes it possible to make use of revisions. Revisions make the models reusable as models with some adjustments can be created by making copies.

2.5.5 Disadvantages of architectural modelling languages

Time
Creating a large amount of architectural models can be time-consuming. The architect must constantly make an estimated '*guess*' and will never know in advance if the number of models constructed will be enough.

Ivory tower
As we have seen previously within the software community, modelling can become an obsession which leads to the creation of a large amount of unused models [Schekkerman]. Often the creation of a large quantity of models also leads to the opposite of architectural awareness and causes organizations to ignore them. This situation is referred to as the 'Ivory tower' and can be avoided by making use of frequent feedback received on the constructed models.

2.6 Business process modelling

2.6.1 Introduction

Business process management concerns the improvement of organization's efficiency and effectiveness by using business and automated solutions. Business process management is visualized by the use business process modelling. Business process models are used to help '*implement the architecture'* as architectural models are only a plan.

During our research we discovered that a task or activity is also referred to as a process. As this is incorrect we will study two definitions to demonstrate the difference.

1. *A business process is a certain activity or set of activities with a specified output, carried out by people and resources in a company. By business process we specifically mean the activity in progress, meaning the control and execution of the activity. Typical business processes are selling a car and fitting a dashboard in a car [Rijsenbrij-i].*

2. *A group of business activities undertaken by an organization in pursuit of a common goal. Typical business processes include receiving orders, marketing services, selling products, delivering services, distributing products, invoicing for services, accounting for money received. A business process usually depends upon several business functions for support, e.g. IT, personnel, accommodation. A business process rarely operates in isolation, i.e. other business processes will depend on it and it will depend on other processes [ITIL-i].*

In definition one we see that Rijsenbrij mainly concentrates on the activity being performed as he specifically mentions the activity in progress being a business process. Rijsenbrij does however mention the use of people and resources performing the activity. Definition two also speaks of activities but clearly mentions that a business process is an activity performed by resources or people and not just the activity itself. We can conclude that a business process is more than only an activity. Further we have determined that the elements of which business processes consists are located throughout the enterprise. Applications e.g. belong to the information

architecture and people belong to the business architecture. Furthermore we conclude that the term process in the business architecture of the DYA architectural framework should be replaced by the term task, activity or business function. A business process cannot be captured within one of the architectural domains as a business process consists of elements belonging to several of the architectural domains.

2.6.2 Objectives

As is the case with the sketches and the architectural models, there are several reasons why architects use this third type of enterprise modelling, they have been summarized below:

Owner identification
Business processes modelling contributes to the breaking down of the business processes. Broken down processes enable organizations to find owners of parts of the business processes [Goedvolk]. When owners of business processes are identified it becomes easier to control the enterprise as the 'key' individuals of an organization, which are important for the success of an organization, can be identified.

Best practices
Business process modelling contributes to finding and to fix processes that are 'best practices' [Van Vliet05]. The best practices can than be copied to another part of the organization or to other organizations. This technique is widely used by organizations that develop large scale information systems like ERP or CRM.

Implementation
Sketches are used to communicate ideas, architectural models are used to work out these ideas and BPM can be used to implement the ideas.

2.6.3 Attributes of business process modelling

Semantics
Not only architectural models but also business process models can be based, partly or entirely, on predefined semantics. When making use of business process modelling one must abide by a large set of rules and guidelines. These semantics of course contribute to the degree of formality of the business process modelling languages.

Specialists
Business process modelling languages usually target the specialist in an organization. The specialist is someone who contributes largely to a process or a number of processes and can help identify and fix the current and required processes.

Operational by nature
Business process modelling is done on a more operational level. The tactic is implemented by using business process models.

2.6.4 Advantages of business process modelling

Guide
Because business process modelling comes with a lot of rules and or guidelines an advantage created is that they can act as a guide for the inexperienced. When

constructing these type of models one requires less 'professional' experience in comparison to the sketches.

Good reflection of reality
As business process models contain a lot of details the description will be a better reflection of reality compared to e.g. the sketches. A good reflection of reality is of importance at this stage as business process modelling concerns the actual realization of the business objectives set out by the strategic dialogue.

Degree of subjectivity is low
As the models contain a lot of detail and they have been constructed in a consistent manner the degree of subjectivity will be low. Users of the models are likely to have the same interpretation of the models. A common understanding is of importance as less communication is required which speeds up the realization.

Complexity of business processes
Today's business processes are very complex. This complexity leads to hardly any insight into the business processes of organizations [Reijers, Aalst03]. Business process modelling enables one to gain insight into the complex business processes as they can be broken down into less complex sub-processes which are easier to comprehend.

2.6.5 Disadvantages of business process modelling

Time consuming
Constructing models that consist of large amounts of details are highly time-consuming. Time is of importance as business objectives need to be achieved to meet market demands. Extensive business process modelling might require too much time so architects must uphold the 'just enough, just in time rule'.

Large quantity of unused models
A common problem with extensive modelling is that large number of models is produced that have no practical use. This is a risk that can become a disadvantage if an architect does not make use of feedback from the business.

Set objective is altered to insure the correctness of the model
A risk that can become a disadvantage is that correct modelling becomes more important then a correct architecture. A problem with business process models is that the large amount of rules and guidelines will not always posses the capability to describe reality in an accurate manner. One must be aware that reality must not be altered to ensure correct models.

Reality differs from simulations
Even though business process modelling contains a large amount of details they are still an abstraction which means that some details are ignored. When simulations seem to meet demands reality might differ.

2.7 Enterprise modelling framework

2.7.1 Introduction

Now we have identified and discussed the use of modelling in general, the use of enterprise modelling and the three types of enterprise modelling we will now proceed by introducing our framework. As we described earlier, models provide an argumentative framework and therefore we will introduce our own model so that our findings can be analyzed. The framework is called the Enterprise Modelling Framework and it will be referred to as the EMF.

2.7.2 Three layers

Our framework, figure 2.4, is a description of enterprise modelling. The framework is based upon the three distinctive types of enterprise modelling. Each distinctive type of enterprise modelling forms a layer of the framework, therefore there are three layers within the framework.

- Layer one: Sketches
- Layer two: Architectural modelling
- Layer three: Business process modelling

2.7.3 Horizontal rule

The horizontal rule is a representation of the time spent on creating the models and the degree of details used in a model as both are directly related. The lower one drops into the EMF the larger the horizontal wideness becomes as do constructing time.

2.7.4 Vertical rule

The vertical rule of the EMF represents two attributes of the enterprise modelling languages. At the top of the EMF the stakeholders targeted, are those with a high ranking position within an organization. At the bottom of the EMF the stakeholders targeted are the people that are lower throughout the organization and these people tend to be more specialists.

2.7.5 Surface

The surface of each layer represents the volume of models produced. Business process modelling will result in a larger volume of models compared to the sketches. This is of course directly related to the architectural nature, as it shifts from strategic nature to a tactical nature to an operational nature, a more detailed description is needed.

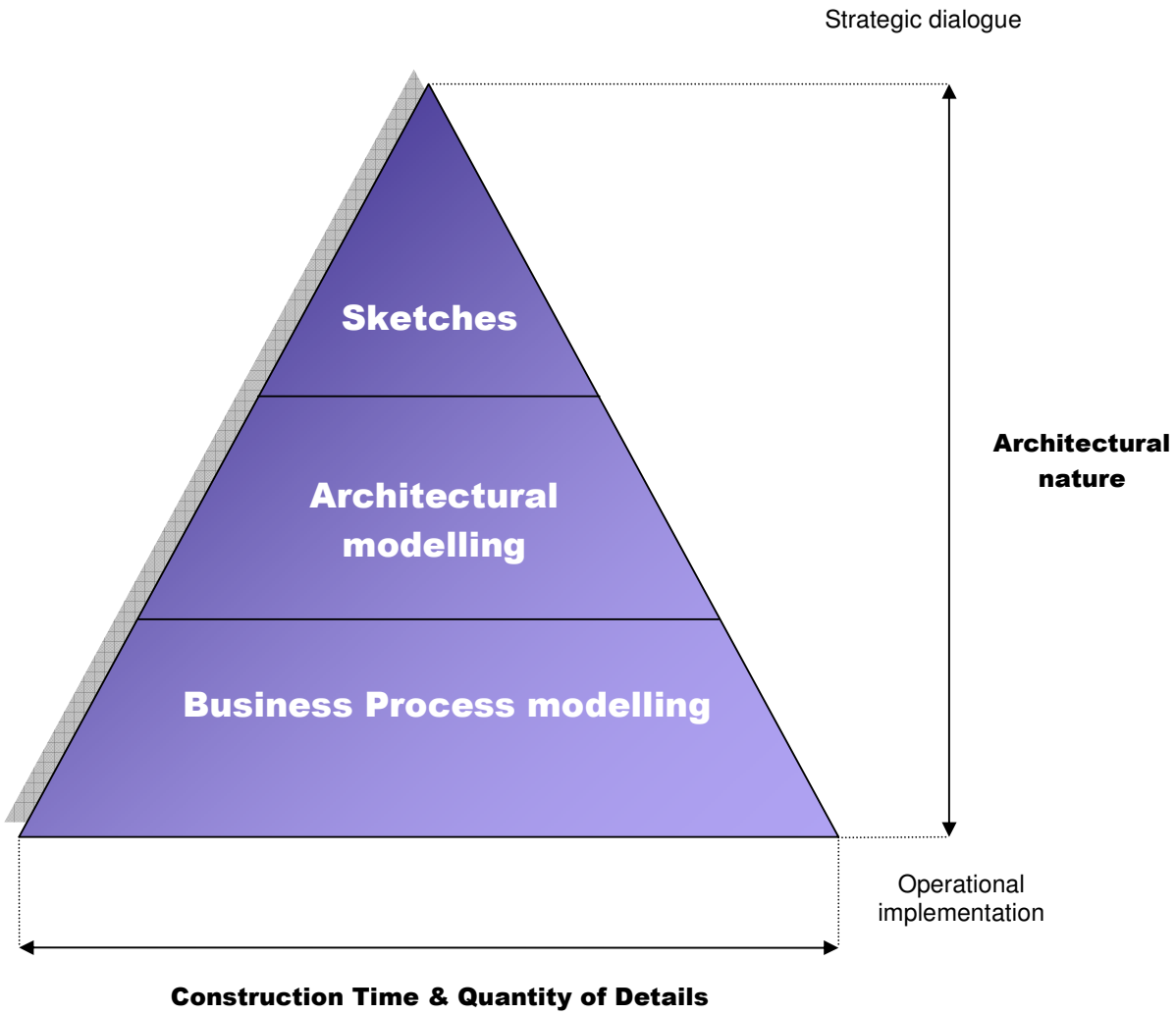

Figure 2.4: Enterprise modelling framework

2.7.6 Why we use the EMF

We now have identified and described the three enterprise modelling types. Secondly we have introduced a framework, which visualizes enterprise modelling that can be used as a discussion point. In the following chapters we will discuss several enterprise modelling languages and position them on the modelling framework.

2.8 Conclusion

We conclude that organizations can benefit from the use of modelling. Enterprise modelling can support organizations becoming more transparent. A transparent organization is one of the elements that an adaptive organization requires. Furthermore we have determined that three types of enterprise modelling exist. Sketches can be used to help determine the strategy for the organization.

Architectural models can be used to help determine the organizational tactic for aligning IT and Business and business process models can help construct the organizational operations.

3 CHAPTHER THREE: MODELLING LANGUAGES

3.1 Introduction

In this chapter we will discuss, judge and compare four enterprise modelling languages: ARIS, UML, IDEF and Archimate. These enterprise modelling languages are the only ones that are capable of modelling a large section of the entire architecture and that are extensively used by organizations. Secondly we will position the modelling languages on the EMF and on the DYA architectural framework so we can compare the different modelling languages. As sketches are defined by an architect a modelling language is not required and therefore, sketches will not be a part of this chapter.

3.2 ARIS: the ARchitecture of Integrated System

3.2.1 Introduction

The Architecture of Integrated System or ARIS was developed by Prof. dr August-Wilhelm Scheer and it was designed not only as an enterprise modelling language but simultaneously as a tool, the ARIS toolset. Scheer has designed ARIS to achieve this objective:

'An integration concept which is derived from a holistic analysis of business processes'

The integration concept Scheer refers to is as a matter of fact the alignment idea. ARIS integration is achieved by identifying the elements of an enterprise at an operational level and seeking their relations as they exist in the business processes. As business results are achieved at an operational level optimized business processes are indeed important and they do contribute to alignment between the architectural domains. It is however so that alignment should not only be sought at only the operational level as business objectives are defined at a strategic level.

3.2.2 ARIS framework

ARIS has its own architectural vision and has translated this into a framework demonstrated in figure 3.1. According to the ARIS method an enterprise consists of five different architectural domains. These domains are represented by four individual views and a combining view. Each individual view represents a part of the architecture and how it is organized. As the four views are separate from each other, a fifth view is used to combine all these individual views, and to demonstrate the overview of the enterprise.

1. Function view
The function view consists of descriptions of performed activities throughout the enterprise. Activities are described as tasks performed upon some object to support one or more organizational objectives.

2. Organization view

The organization view describes the structure of an organization. This view forms a description of the entities within an organization that perform the tasks in the function view.

3. Data view

The data view describes the data objects throughout the organization.

4. Product / service view

The product / service view is used to describe the product or service that an organization offers. As we just described the activities of the function view are performed to support organizational objectives which are producing a product or service. If the organizational objectives contribute to the success of the organization they are considered of value to the organization. This is why a product or service is referred to as the result of an activity out of the function view which offers value to the organization.

5. Control view

The control view is the view used to create an overview between views one to four. As just described there are several relations between the views one to four. These relations are described in the control view.

The ARIS framework mainly concentrates on the 'business' side of the enterprise as the organization view, function view and the product/ service view are parts of the business. The data view demonstrates the importance of data to the organization but as most of it is stored and structured through the use of (information) technology we wonder why information technology was not given a place in the ARIS framework. The control view demonstrates how ARIS wishes to encompasses the alignment between the four individual views. Alignment is gained by relating the four individual views in business processes which can be controlled.

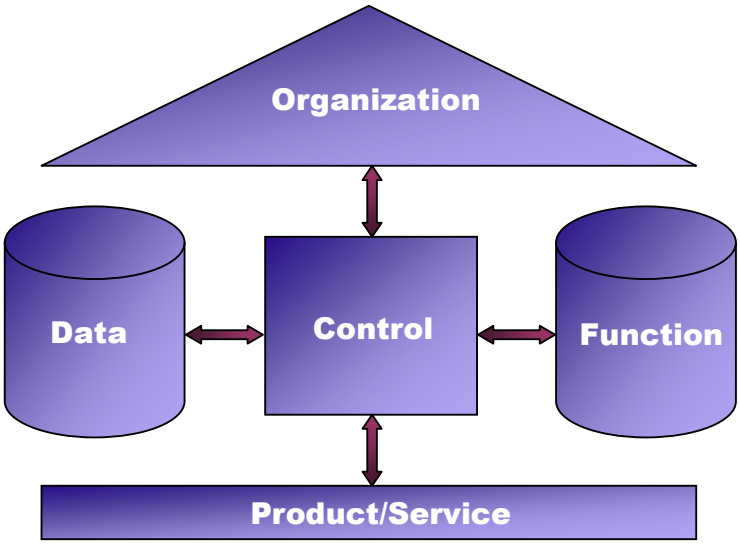

Figure 3.1: ARIS architectural framework:

3.2.3 ARIS main models

Each ARIS view is represented by a number of different models. In this section we will discuss these models.

Organizational objectives & strategy
The balanced scorecard is used to create descriptions for strategic scenarios and helps create key performance indicators so that it is possible to perform measurements. The cause and effect diagram can be used to model the organizational objectives and the strategy to achieve these objectives [IDS]. The objectives and strategy are modelled in such a way that several perspectives are offered in the form of swim lanes. These perspectives enable a user of the models to determine which stakeholder, e.g. a client or supplier, is related to a specific organizational objective. This is demonstrated in figure 3.2 [IDS].

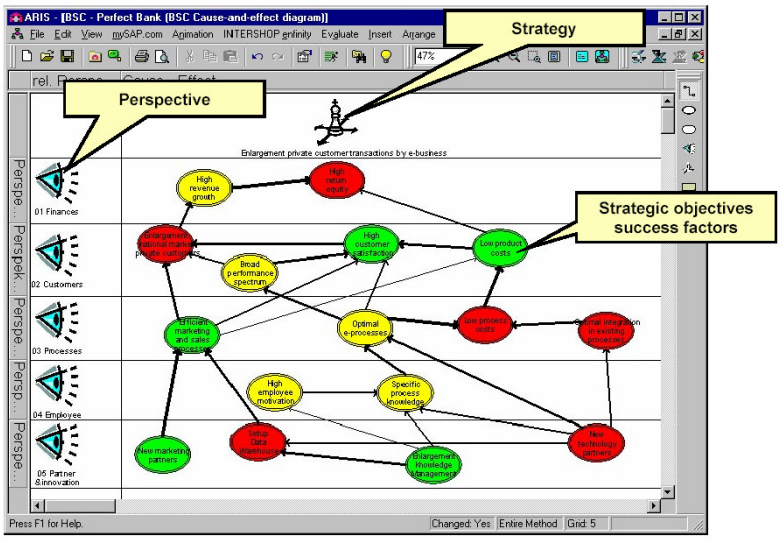

Figure 3.2: Cause and effect diagram

Value added chain diagram
The value chain distinguishes the primary business functions, and their sequence, from the support functions. The primary functions are those that are directly related to the organizations value. The value added chain diagram describes which organizational functions, as described in the function view, are of direct value to the organization and to which organizational objective they contribute. The diagram is demonstrated in figure 3.3 [Spiekerman03].

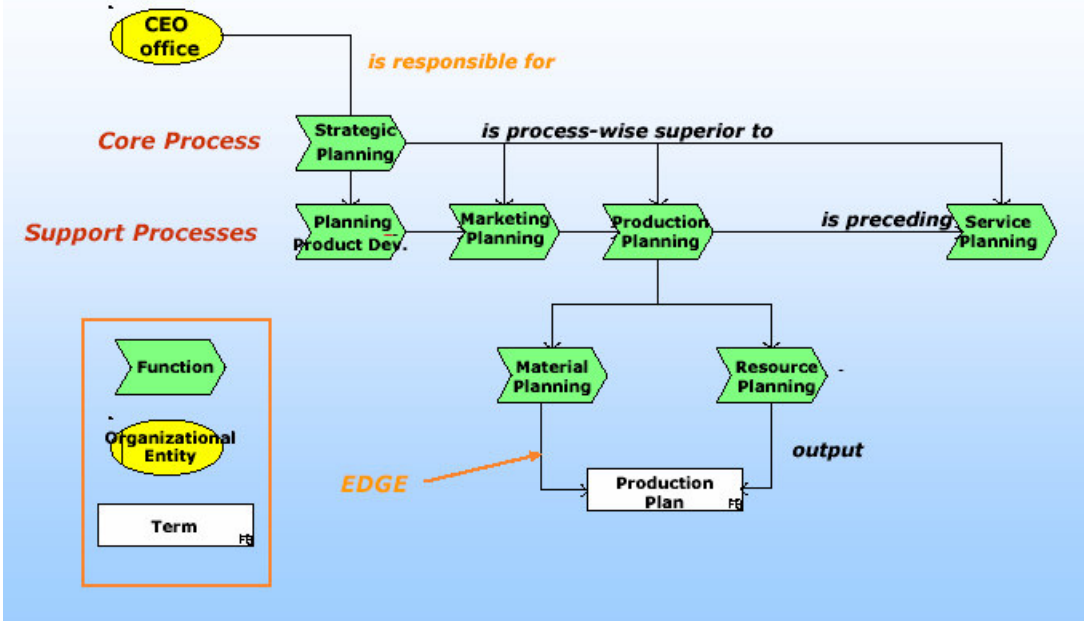

Figure 3.3: Value added chain diagram

Organization structure

ARIS makes use of the organization structure diagram to describe the static structure of an organization. The organization structure consists of organization units, which are the performers of the functions that must be performed in order to achieve some set objective, and positions which are the smallest organizational units. An organization unit can e.g. be an organization division. A position can be assigned to a person who is the smallest unit. Structure is achieved through decomposition. This means that a hierarchal structure is created with at the top the highest organization unit, the organization itself, and if required at the bottom the various people(figure 3.4).

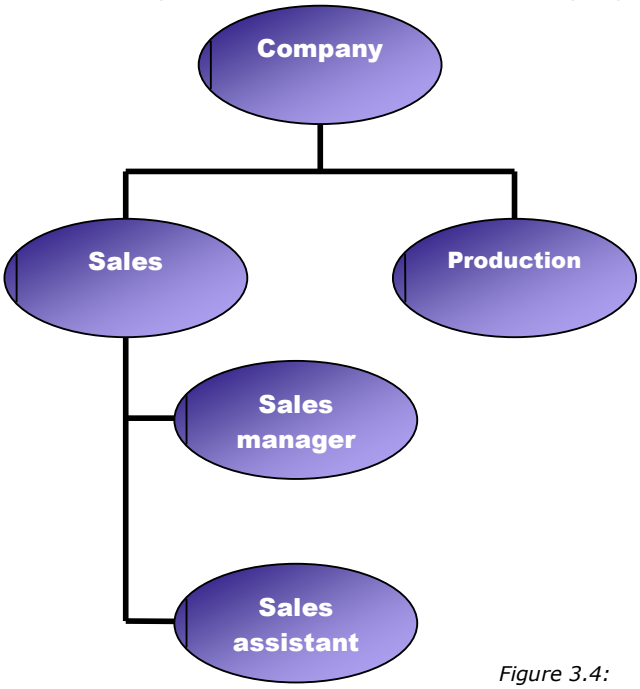

Figure 3.4: Organization structure

Function tree

The function tree describes the tasks that are performed throughout the organization. The function tree is also a decomposition diagram but now of a task. Each task is decomposed so that a hierarchical tree forms (figure 3.5).

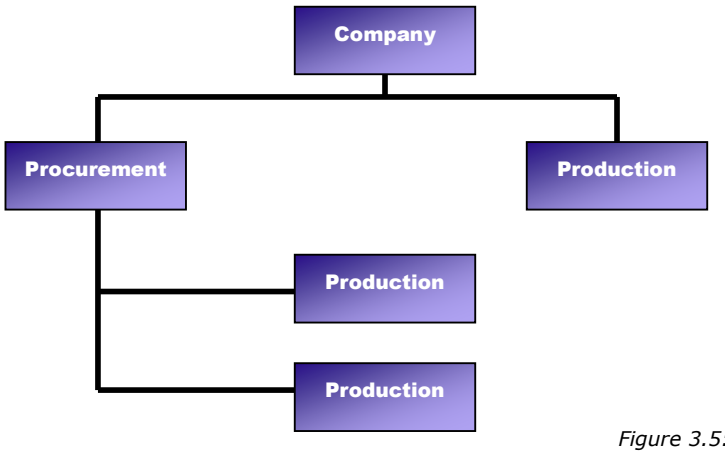

Figure 3.5: Function tree

Data modelling

Data modelling can be performed in ARIS by making use of two different modelling techniques, Entity relationships diagram(ERD) and document type definition(DTD). ERD's are used to define data objects, also referred to as business objects, and their relations within an organization. An ERD is actually a high-level description of conceptual data. DTD's can be used to define the structure of information within the organization.

EPC: Event Driven Process Chain

The event driven process chain (EPC) describes the sequence in which the functions, of the function tree, are performed along with it's relation to the organization units (organization structure) and the data objects (ERD or DTD) as demonstrated in figure 3.6 [Vernadat]. The previous models indicate *'of what an organization consists'*, the EPC's describes *'how the organization is arranged*.

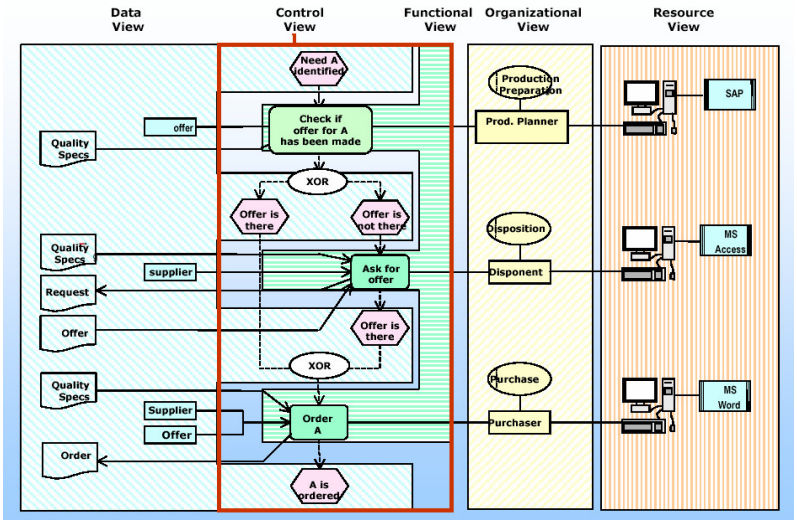

Figure 3.6: Event driven process chain

3.3 UML: Unified Modelling Language

3.3.1 Introduction

The unified modelling language, or UML, unifies a set of methods that have mainly originated from Grady Booch (Booch Method), James Rumbaugh (OMT) and Ivar Jacobsen (OOSE). UML was later accepted as the standard for object oriented modelling by the OMG. UML is mainly used for software systems development but today it is also used to model parts of the business.

3.3.2 UML framework

The UML Framework, figure 3.7, is called the 4+1 model. It consists of four separate views and one combining view. Each view represents a specific aspect of an (information) system. Each view can be used individually to satisfy a number of stakeholders.

1. Design view
The design view describes the structure of a system in terms of classes and objects that exists within a system.

2. Process view
The process view describes the dynamic behaviour of a system in terms of the states a system can take on and the transitions between the several states.

3. Implementation view
This view describes how the components and files are structured to form a physical system.

4. Deployment view
The deployment view is to demonstrate how the entire system is distributed and installed. It describes the relation between the software and hardware.

5. Use case view
The use case view describes the behaviour of a system. It does not offer insight into the structure of a system but how it behaves as seen by a user of the system.

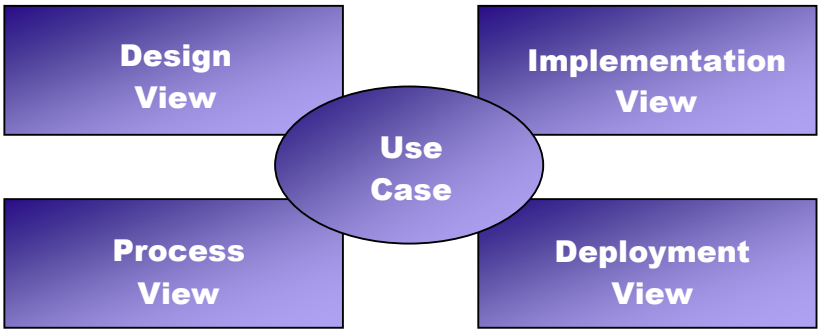

Figure 3.7: UML Architectural framework

The design view and process view focuses primarily on the software and the deployment view and implementation view focus primarily on the hardware aspects of information technology. UML combines all separate views from a user's perspective in the use case view. By doing so the information technology is designed from a user's perspective. UML states that user demands are primarily business driven because they wish to accomplish some set business objective. It is however so that the direct relation between the business objectives and the information technology is unclear. There is no view that enables us to determine any relationship between the business objectives and information technology as we only see that the information technology is related to user demands. When UML is used one must trust that the user demands have truly originated from a set of business objectives.

3.3.3 UML models

Use case diagram
A use case is a written description of an interaction between a system and an actor. A use case diagram is a graphical model of a use case (figure 3.11).

State diagram
The state diagram describes the several states which a system can take on and the relationships between them.

Activity diagrams
Activity diagrams are a specialized form of state diagrams and they are used to describe the workflow or the so called process flow belonging to a system. Activities, actions, transitions and states can be elements of an activity diagram (figure 3.11).

Class diagram
The class diagram describes a system on a more detailed level. Objects and their relationships are the elements of a class diagram. The class diagrams are mostly used for data modelling and detailed software design.

Component diagram
The component diagram is used to describe how the system is constructed (figure 3.8). A component diagram describes the components and their relations of which a system consists. These types of diagrams are usually used in larger teams where component based software development is used.

Deployment
Deployment diagrams are used to describe the system during run-time. This means that the deployment diagrams are used to model the hardware, software and middleware as it is used in real-time (figure 3.8).

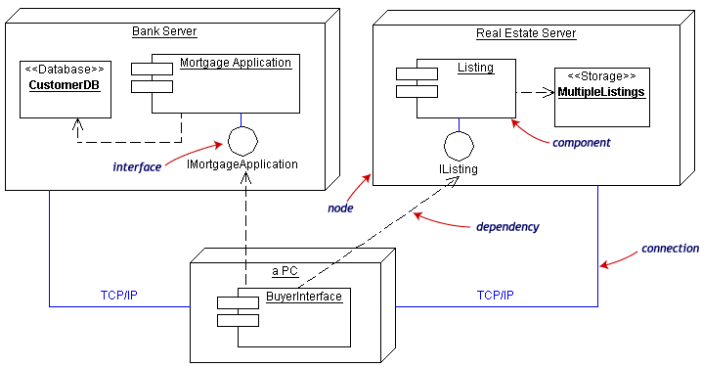

Figure 3.8: Component & Deployment diagram

3.3.4 UML business extensions

As we have previously described, UML was primarily created for the development of software systems. In time however, it has been applied for modelling the enterprise. As UML is not designed for modelling the entire enterprise a number of extensions have been introduced by [Errison, Penker], [Agile alliance] and [Pan-Wei] to make UML a more usable enterprise modelling language instead of only software system development.

The Errison-Penker UML business extensions are the most developed extensions. They foresee in a new arrangement of the original UML 4+1 framework and have adjusted some of the modelling methods.

1. Business process view
This view describes the functions that offer value to the business. The main objective of this view is to describe the relationships between the different business functions. As with DYA we believe the term process should be replaced by the term business function as not the processes but the tasks are described.

2. Business structure view
The structure view is used to describe the organization structure and the product and or services that an enterprise offers.

3. Business behaviour view
The behaviour view describes the behaviour of the resources and processes of the enterprise.

4. Business view
This view covers the objectives set by an enterprise and the strategy to reach these objectives.

The views of the UML business extension focus primarily on the business side of the enterprise where the original UML views primarily focuses on the information technology side of the business. If both the original UML and the UML business extensions are applied both the business and information technology can be described. A downside is that the UML business extensions have no clear relation to the original UML even though the business extensions have originated from UML. Combined it is possible do describe the business objectives, the user demands and the information technology but a clear relation between the business objectives and the user demands is still not available. A model that relates the origin of the information technology to the business objectives would be a useful extension as it would enable us to determine if the information technology still meets the business objectives.

3.3.5 UML business extensions models

Goal model
The goal model is used to describe the objectives set by the enterprise. The goal model is based on the UML class diagram. Each objective is modelled together with its sub-objectives. Each objective is assigned with a current situation and the desired target.

Organizational model
The organization model offers a description of the organization structure and two types of models have been introduced to make this description. The first is a model

based on the UML class diagram. The second diagram (figure 3.9), the mostly used diagram, is a hierarchal tree of the structure of the organization and it is similar to the ARIS diagram.

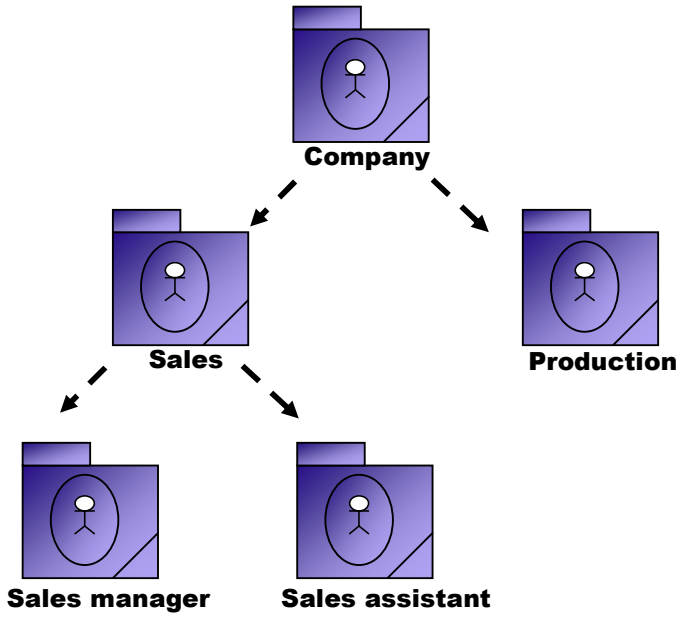

Figure 3.9: Organization model

Process hierarchy model

The process hierarchy diagram has been introduced to describe the hierarchy of the organization functions. The model consists of functions of an organization that have been decomposed as demonstrated in figure 3.10 [Agile alliance].

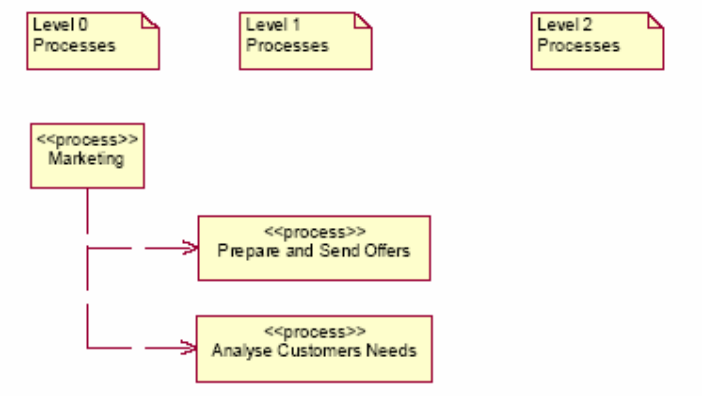

Figure 3.10: Process hierarchy model

Use-Case diagrams for business modelling

[Pan Wei] has introduced the use cases for business modelling to describe the relation between the business actors, which are the users or people throughout organizations, and the business functions. This model is demonstrated in figure 3.11 [Pan-Wei]. A use-case model is then created by making use of the standard UML case-diagram technique but with symbols introduced in the extension.

Updated Activity Diagram
The activity diagram is used to describe the process flow within an organization. The sequences in which the functions, as described in the process hierarchy, occur are modelled. Furthermore they describe which business actor performs the function and which business objective is the result of a task. It is also possible to include the resources that contribute to the process. The model is demonstrated in figure 3.11 [Agile alliance].

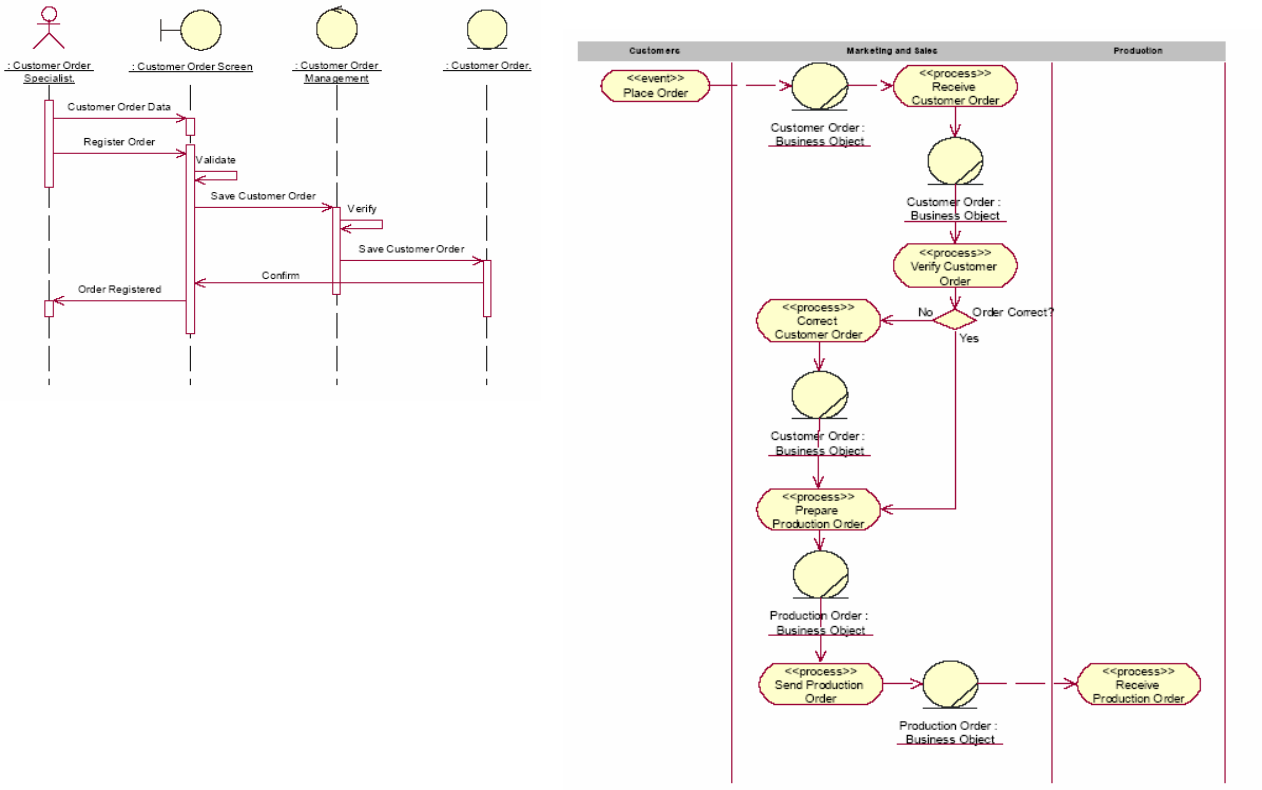

Figure 3.11: Use case diagram and activity digram

3.4 IDEF: Integrated computer aided manufacturing DEFinition language

3.4.1 Introduction

The integrated computer aided manufacturing definition language, or IDEF, was developed by the US Air Force and is a set of fifteen different modelling methods. IDEF was one of the first modelling languages for system development and it does not posses any framework

3.4.2 IDEF Models

IDEFØ: Function modelling
IDEFØ models can be created that describe the decisions and activities of a system. The IDEFØ models disregard sequence and time. IDEFØ shows *'the elements of which a system consist' 'not what a systems does'* or *'how the system does this'.*
The IDEFØ models are mainly used to model functions of a computer system and those belonging to organizations and usually consist of an AS-IS and a TO-BE situation. After all the IDEFØ models are created, a node tree model is introduced which is a hierarchal tree that describes the decomposed functions. This model is demonstrated in figure 3.12 [IDEF].

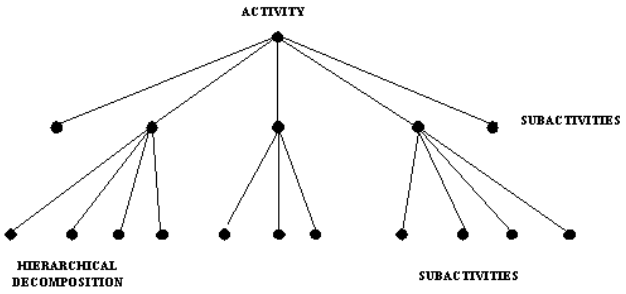

Figure 3.12: IDEFØ node tree

IDEF1: Information modelling
The IDEF1 method is used to describe the information on the objects that exists in an enterprise. IDEF1 only distinguishes concepts of information, and their relations, within an organization. Such a conceptual view helps organizations to understand which information exists and how it is handled. An IDEF1 diagram is demonstrated in figure 3.13 [IDEF].

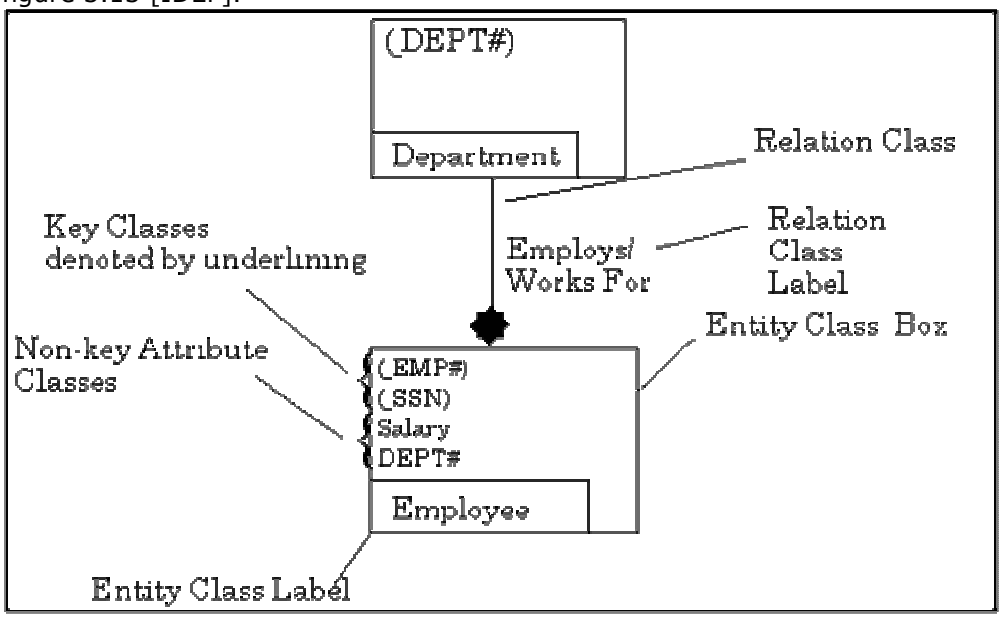

Figure 3.13: IDEF1 diagram

IDEF 3: Process description capture
The IDEF 3 method is used to describe the processes of a system. The IDEF3 method itself consists of two different modelling techniques. The first is the so called process description diagram which captures the processes and their relations within the context they occur. The second IDEF3 modelling technique is the object state transition diagram. This diagram makes use of the processes out of the process description diagram and describes the transitions between these processes. Together they describe '*how things work*. An IDEF3 diagram is demonstrated in figure 3.14.

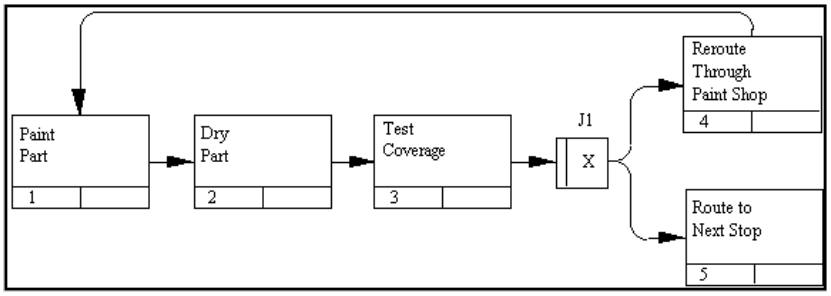

Figure 3.14: IDEF3 diagram

IDEF: Remaining models
The IDEF2 method for simulation modelling and the IDEF4 method for Object-Oriented design are not used for enterprise modelling. The IDEF5 till IDEF14 methods do exist but could be seen as methods under construction. These methods have been academically explored by scientists but have never been used. Only IDEF12 for organization modelling seems to be of use for enterprise modelling but there is little information available on this method.

IDEF is a gathering of several methods used to describe several aspects of both information technology and processes. By doing so IDEF has resulted in a modelling language with the capability to describe several aspects of an organization but that there is no relation between the descriptions. An explanation can be found in the time that IDEF was defined. An overall approach for constructing descriptions did not yet exists but separate methods had been developed. Instead of developing one new modelling language separate methods were combined to form IDEF to meet user demands. One can argue that it would have been better to develop a new modelling language or that IDEF should have been adapted over time but as user demands differed at the time this is clearly of no use.

3.5 Archimate

3.5.1 Introduction

Archimate is an enterprise modelling language that has been developed by the Telematica Institute in cooperation with the Radboud University of Nijmegen, ABN-AMRO, the Dutch tax-office, the ABP, the LIACS, the CWI and Ordina [Lankhorst05].

3.5.2 *Archimate Framework*

The Archimate framework, figure 3.15, consists of three distinctive layers of architecture and three aspects of architecture. The architectural domains according to Archimate are projected on the framework. [Lankhorst04] has provided us with the following description of the Archimate framework layers.

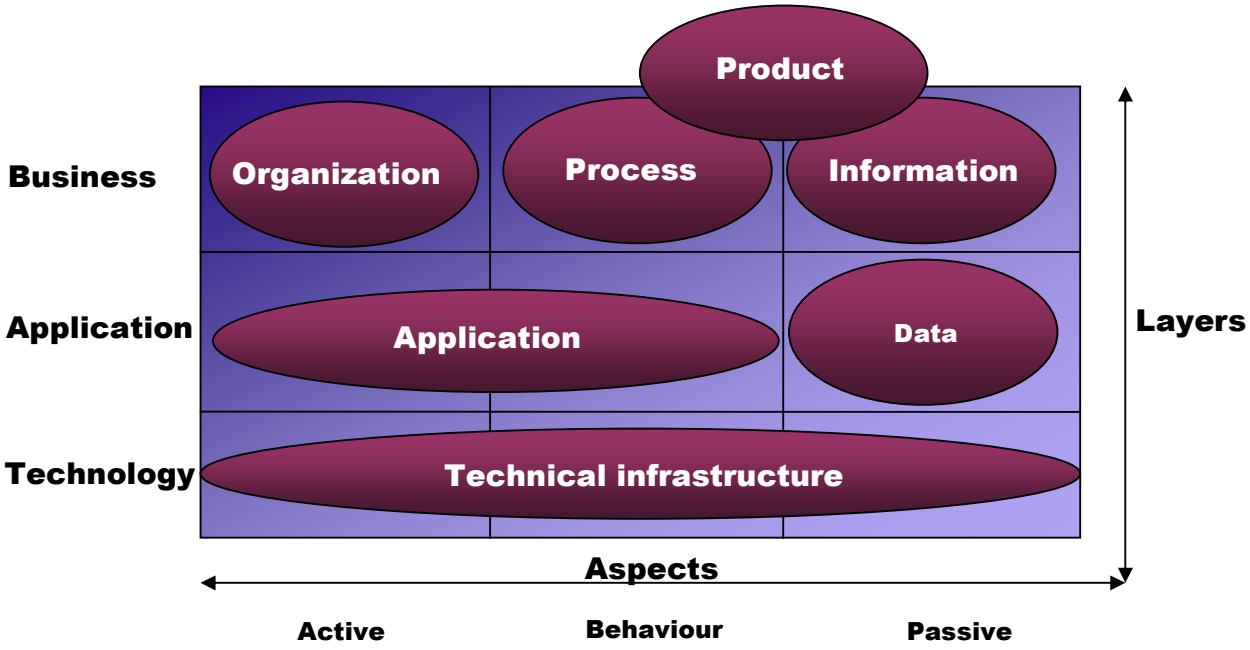

Figure 3.15: Archimate Architectural framework

1. Business layer
The business layer describes the products and services that are offered to external customers. The products and services are outcomes of the business processes.

2. Application layer
The application layer supports the business layer with application services which are realised by (software) applications.

3. Technology layer
The technology layer offers infrastructural services (e.g., processing, storage and communication services) needed to run applications, realised by computer and communication hardware and system software.

Furthermore we see that the framework is also divided into several aspects. The aspects range from active to passive. The passive aspect is the section of the organization that undergoes actions performed by the passive aspect through the behaviour aspect.

3.5.3 *Archimate view & viewpoints*

Archimate offers the capability to define views. An architect can choose a viewpoint by making use of the Archimate framework. This capability enables the architect to define

a view and thus a model that is 'fit' for a certain purpose which is usually directly related to the needs of the stakeholders.

3.5.4 Archimate models

Archimate focuses on views, which can be a model, and viewpoints. Because the viewpoints can be defined according to the stakeholders requests the number of models that can be created to describe the enterprise can be enormous. Furthermore the Archimate project team has predefined several viewpoints and views. Instead of discussing several models we will demonstrate several viewpoints that are positioned on the Archimate framework.

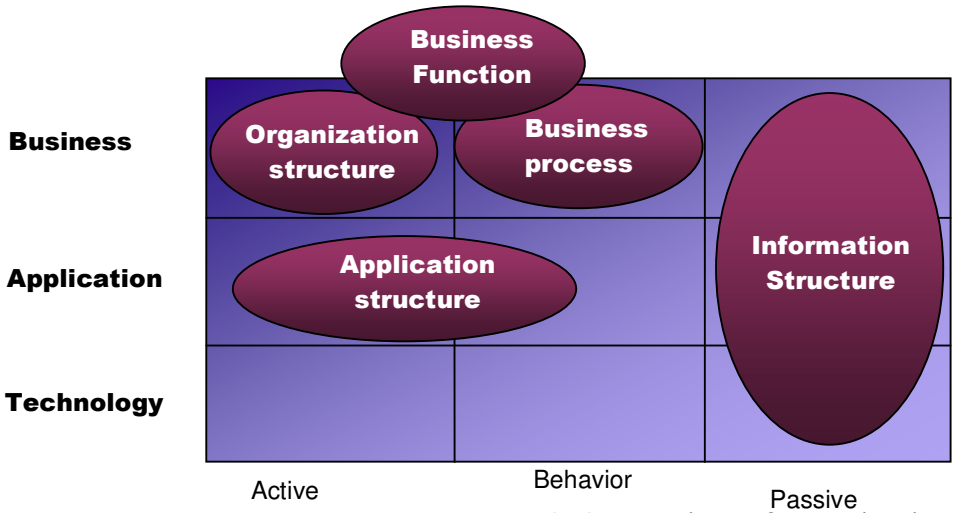

Figure 3.16: Archimate framework and several viewpoints

In figure 3.16 we see that a viewpoint business function exists. This viewpoint will describe the business functions that exist in the organization. The view business process will offer a description of the business processes. In the figure we also see that the organization structure (active aspect) will perform adjustments on the information structure (passive aspect) through the use of business processes (behaviour aspect).

Archimate is a modelling language that has clearly been developed around today's demands. All aspects of the enterprise have been given a place in the Archimate framework and the relationship between the aspects can be defined by choosing a viewpoint. A remark we can make is that there is more attention for the technology and less attention for the information which together form the information technology. Two of the three layers consist of technology: Application and technology.

In chapter one we described information actually being data structured by the use of applications. In figure 3.15 we can see that the Archimate vision of data corresponds to that of our own as the data is a domain belonging to the application layer.

In figure 3.16 we also see that Archimate clearly distinguishes business functions, being activities or tasks, from a business process. This is in line with the comment we have made in paragraph 2.6 that an activity or task is not the same as a process.

In contrary to the DYA architectural framework the Archimate architectural domains have been distinguished over several aspects. A domain can e.g. be passive or active. Especially for a dynamic architecture this can be of value as mostly the active

domains, e.g. the organization structure, will change fast as they are driven by business demands. A passive domain, e.g. data, will not adapt on its own and a passive domain needs to be constantly synchronized as it will eventually 'slow down' the enterprise.

3.6 Other modelling methods

A lot of modelling methods exists that can be used to describe sections of organizations which are also commonly used. If we look at process modelling Petri nets or flowcharts are commonly used to describe processes. It is however so that these type of models can only describe a part of the organization. It is of course possible to gather several other modelling methods and combine them in such a way that it is possible to describe the entire or at least a large section of an organization. A problem that arises is that sections of the organization will be described double or not at all. This is because no true relations exist between the modelling methods. A benefit of a modelling language is that the set of modelling methods have been chosen or designed in such a way that they are in line with each other.

3.7 Modelling Languages & the EMF

3.7.1 Introduction

Now we have described the several enterprise modelling languages we wish to determine their relationship with the layers of enterprise modelling. Secondly we wish to compare the enterprise modelling languages to each other.

3.7.2 Positioning on the EMF

IDEF can be used in several areas. As we know it has first been developed for software engineering. Secondly it is used in the field of Business Process modelling. UML is also used in several areas but as with IDEF its original primary goal is software engineering. The UML extensions make it possible to perform in the area of business process modelling. Because both IDEF and UML can be used for system engineering we have visualized this concept in figure 3.17. We have positioned both UML and IDEF between the third layer of the EMF and the system engineering concept to demonstrate that they can be used for both.

ARIS was created for Business Process modelling. If we look at the ARIS framework we can determine that the control view is the centre of the framework. The control view primary addresses processes as we have previously discussed. We have positioned ARIS on the business process modelling layer of the EMF in figure 3.17. According to my opinion Archimate is the only real architectural modelling language that exists nowadays and is therefore positioned it on the second layer of the EMF.

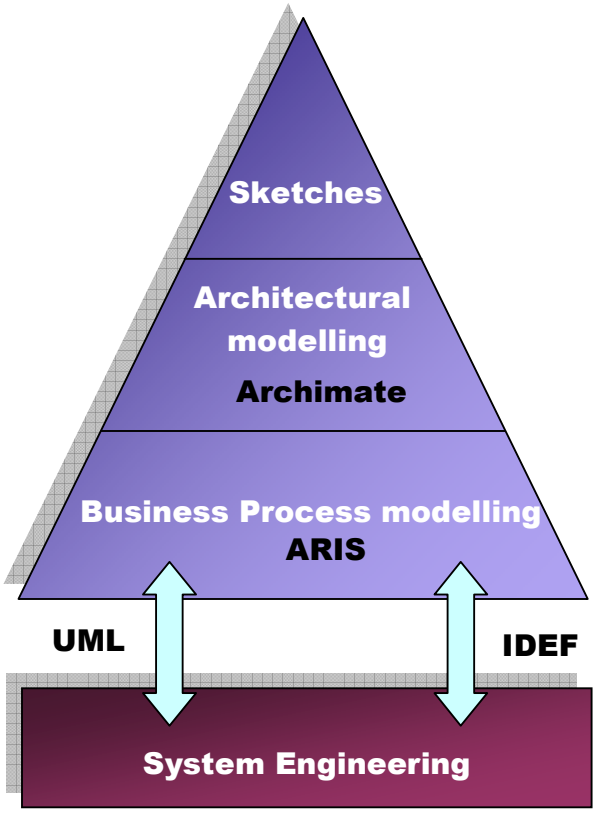

Figure 3.17: Modelling languages positioned of the EMF

3.7.3 Positioning on the DYA framework

As Archimate is a different type of modelling language, there is no point in comparing it to other modelling languages. ARIS, UML and IDEF are all modelling languages belonging to the third layer of enterprise modelling and we will attempt to compare them to each other. By making use of the DYA architectural framework we have a basis for comparison. We have positioned the discussed methods on the DYA architectural framework in figure 3.18. Actually we only use the third layer of DYA, the models, and dividing it over the three business process modelling languages.

	Business goals							
	Business architecture			Information architecture		Technical architecture		
	Prod/ Service	Organiza-tion	Process	Data	Application	Middle-ware	Platform	Network
ARIS		Organization structure & Value chain & Cause/effect	Function tree & EPC	ERD & DTD				
UML		Organization Model& Goal Model	Use-Case & Activity diagram	Class Diagram	Deployment Diagram			
IDEF		IDEF 12	IDEF0 & IDEF3	IDEF1				

Figure 3.18: DYA architectural framework and the modelling languages

Organization modelling
The ARIS organization structure diagram and the UML organization model both have the same objectives and the same structure. As both make use of hierarchal decomposition and both have no further modelling rules we can conclude that both diagrams are the same and only their symbols differ (see figure 3.4 & figure 3.9). We can make no comparison to the IDEF12 method for organization modelling as little information is available.

The UML goal model and the ARIS cause and effect diagram bear some similarities as they both have as objective describing the organizational objectives. ARIS and UML do however differ in their approach for providing a description of these models. ARIS is also the only modelling language in which it is possible to model the organizations strategy. A comparison with IDEF is not possible as it does not provide the capability to model the organizations objectives and strategy.

Process modelling
As we see, in figure 3.18, all modelling languages posses the capability to model the processes of an organization. The ARIS function tree and the UML Process hierarchy models both have the same objective. The rules on modelling are the same and only their symbols differ. The IDEFØ diagram differs in the description it offers on the organizations function. The node tree, that is a result of the IDEFØ diagram, does offer the same description as the ARIS function tree and the UML Process hierarchy model, only the symbols differ.

All modelling languages have the capability to describe the process flow within an organization. The description does however differ. ARIS makes use of the EPC, UML of

the Activity diagram and use-cases, and IDEF3 for process modelling makes use of the process description diagram and the object state transition diagram. Another difference is that both UML and ARIS encompass a larger part of the enterprise into their business processes compared to IDEF.

Application & Information modelling
UML makes application modelling possible through the deployment diagram. IDEF does not foresee in a method designed for application modelling although the IDEFØ diagram can be used for this purpose. ARIS involves the applications in the control view or the data view.

The ARIS ERD, the UML class diagram and the IDEF1 method can also be used to model information. The objective is the same and there are a lot of similarities. Information objects and their relations are defined in much the same way but we cannot say they are identical.

3.8 Enterprise modelling criteria

3.8.1 Relevance of the criteria

Enterprise modelling criteria are useful as the non existence thereof contributes to the lack of insight architects have into enterprise modelling. In the current situation architects are unable to determine if a modelling language will foresee in their needs. The criteria will also enable the architects to make a better choice for an enterprise modelling language.

3.8.2 Set of criteria

Architecture involves several stakeholders and they come in many different types and numbers. Defining the stakeholders depends on an architectural vision, an organization or even an individual architect. There is no way to determine if an enterprise modelling language offers the possibilities to satisfy the needs of all stakeholders in an organization as the stakeholders themselves are undefined. It is however possible to determine if an enterprise modelling language has acknowledged the existence of stakeholders and their importance to architecture and because architecture revolves around the organization stakeholders we feel a criterion is required to determine this.

'Criterion: Enterprise modelling languages must recognize the existence of stakeholders and recognize their importance to architecture'

In [Rijsenbrij05] the concept of a view and a viewpoint are discussed. A viewpoint is described as an approach to a 'system'. Such an approach results in a view. When determining a viewpoint one must ensure that the resulting view is of use to a stakeholder. Enterprise modelling languages are used to create models which can actually be considered a view. Views are constructed for stakeholders and we feel such a view must foresee in the needs of the stakeholders. This leads to the following criterion:

'Criterion: The resulting views of an enterprise modelling language must foresee in the need of the stakeholders'

Analysis and discussion are not only performed by the architect but also by the other stakeholders. The other stakeholders must be able to interpret and understand the description that is given [Aarssen]. The description that the models offer must contain symbols that a stakeholder can understand otherwise they are of course useless. This argument is also supported by [Giga03] which mentions that business executives are not interested in models as they mostly do not understand them.

'Criterion: Stakeholders must be able to understand and interpret the descriptions offered by an enterprise modelling language'

A common argument that is given, in the interviews with [Proper], [Hoppenbrouwers], [Berg] and [Broekema], is that methods must somehow meet standards already set. We agree because standardization offers the possibility of easy shift from one to another modelling language and that different types of modelling languages can also be combined. We have defined the following the criterion:

'Criterion: Enterprise modelling languages must adhere to some kind of standard'

As we have discussed, in chapter two, there is a need for transferable enterprise models. Models that can be transferred between architects are useful because they do not require the knowledge and or presence of the person who has created the models. We have defined the following criterion to meet this demand:

'Criterion: Enterprise modelling languages must posses transferable models'

Knowledge of the environment in which an enterprise functions is necessary. The environment of the enterprise (clients, suppliers, government, etc.) is of influence to the way in which an enterprise functions. Therefore not only the enterprise but also the environment in which the enterprise functions must be modelled.

'Criterion: Enterprise modelling language must offer the capability to model the environment in which the enterprise functions .'

If an organization requires a full insight, it needs to have a model of the entire enterprise. Even though it is not always necessary to model the entire enterprise, it should be possible to do so.

'Criterion: Enterprise modelling language must be able to model the complete enterprise.'

In [HP03-2] alignment of business goals and IT strategy is seen as critical for the success of an organization. Alignment between IT and the business is of importance as the business relies on IT. As architecture not only covers the enterprise and its domain but also the alignment between the domains, alignment must also be part of an enterprise modelling language.

'Criterion: Enterprise modelling languages must be able to model the alignment between the architectural domains.'

Decisions made throughout an enterprise must be based on solid data. A modelling language that is still being developed could compromise the correctness of the information. Therefore the enterprise modelling language must be mature.

'Criterion: an enterprise modelling language must be marked as fully developed by its creators.'

The construction of an enterprise modelling language can be a time consuming activity. When making use of a modelling language an architect must be able to work efficiently. An enterprise modelling method must have the capability to have an appropriate performance, relative to the amount of resources used, under stated conditions. In practice this means that a tool is necessary that can be used for modelling which is supported by [Schekkerman] and [Berg].

'Criterion: For the creation and maintenance of models to be efficient, a tool must exist'

3.9 Quality of the Enterprise modelling languages

3.9.1 Introduction

In this section we will apply our introduced criteria on the discussed enterprise modelling languages to provide insight into the quality of the enterprise modelling languages.

3.9.2 Criteria applied

Criterion: Enterprise modelling languages must recognize the existence of stakeholders and recognize their importance to architecture
Each UML view is described by making use of several UML modelling methods. The several UML views have been defined to offer more than one approach to a system as UML user demands may differ. UML mainly offers descriptions for stakeholders which are part of the information architecture and the technical architecture.

ARIS has not defined any stakeholders but as with UML, it has defined several views which can be described with the use of several models. ARIS mainly concentrates on

stakeholders that have an interest in the organizational and process aspects of the business architecture.

IDEF has not defined views or stakeholders. The concept of an architectural stakeholder was unknown at the time IDEF was designed. We can say that the more technical stakeholders can benefit from IDEF and that IDEF doesn't posses any models designed for business stakeholders.

The stakeholder concept is recognized by Archimate as it has actually been developed around the concept of a stakeholder. The viewpoints and the resulting views are defined to meet the demands of defined stakeholders.

Criterion: The resulting views of an enterprise modelling language must foresee in the need of the stakeholders
The ARIS and UML viewpoints, and the IDEF models, have been defined by the developers of the modelling languages. As each organization will have its own unique stakeholders and stakeholder demands, a consequence of defining all the viewpoints in advance, is that the stakeholder demands cannot always be satisfied. Even though Archimate has defined several stakeholders and viewpoints, it is possible to create more, which makes Archimate more flexible than the other modelling languages.

Criterion: Stakeholders must be able to understand and interpret the descriptions offered by an enterprise modelling language
It is sadly not possible to measure the degree of stakeholder satisfaction before the models have been constructed. We can however determine the likelihood that a stakeholder will be able to understand and interpret a modelling language.
IDEF and UML have originated from system development and they are likely to be understood by stakeholders that are involved in the information architecture and technical architecture. The UML business extensions still make use of a lot of rules and symbols of the original UML and are therefore unlikely to be understood by the business oriented stakeholders. There are no IDEF business extensions and the original methods were created for system developers, it is therefore unlikely that they will be understood by the more business oriented stakeholders.

ARIS focuses mainly on the business architecture and it makes use of symbols common in this area and it is therefore likely that stakeholders involved in the business architecture will understand the models that have no experience with ARIS.

The Archimate models are likely to be understood by stakeholders for two reasons. The first reason is that the viewpoints are constructed around the stakeholders; the second reason is that the symbols are commonly used.

Criterion: Enterprise modelling languages must adhere to some kind of standard
UML is actually a standard for system development [OMG], so it adheres at least its own standard as do the UML business extensions. Furthermore the ARIS ERD and DTD are all standards for information modelling. Archimate is partly based on UML and it is also designed to 'co-exist' with UML [Hoppenbrouwers]. IDEF itself is also a standard [NIST93].

Criterion: Enterprise modelling languages must posses transferable models
The UML and IDEF models are highly transferable in the area of information architecture and technical architecture as they are both standards in this area. The ARIS models for the business architecture can be transferred between different architects as they posses a great deal of detail which makes them less subjective.

Furthermore the symbols are commonly used. The Archimate models are consistent and can be transferred between architects that are acquainted with the modelling language. This is also supported by [Proper].

Criterion: Enterprise modelling language must offer the capability to model the environment in which the enterprise functions
ARIS, UML and IDEF hardly posses capabilities to model the external environment and focus mostly on the internal organization. In [Lankhorst04] the external environment is discussed and it can be described as part of the business layer or a fourth layer, the environment layer, can be added to the Archimate framework.

Criterion: Enterprise modelling languages must be able to model the complete enterprise
In figure 3.18 we see that ARIS, UML and IDEF don't or hardly offer the possibility to model the product / service architectural object. ARIS, UML and IDEF all offer the possibility to model the organizational and process architectural object. With UML it is also possible to model the application architectural object. Finally we see that UML, ARIS and IDEF are all able to construct models that describe the data architecture.

It is possible to construct models of the entire Archimate architectural framework. As there is similarity between the DYA architectural domains and the Archimate architectural layers, both exists of business and technology, we conclude that it is possible to model the entire architecture.

Criterion: Enterprise modelling languages must be able to model the alignment between the architectural domains
IDEF has no architectural domains so alignment cannot be achieved. In both ARIS and UML, alignment between the views is achieved by combining them in an overall view. Archimate offers much more possibilities to describe the alignment between the architectural domains. The architect can choose a viewpoint which consists of one or more architectural domains.

Criterion: An enterprise modelling language must be marked as fully developed by its creators.
Although the ARIS modelling language is constantly being improved, the discussed ARIS methods have passed the development stage and are now evolving as time goes on and demands change. IDEF has been operational for a longer period of time but is no longer evolving and a consequence for IDEF is that over time it will no longer meet the market demands. The developer and main user of IDEF, the US Air force, is currently seeking a replacement [Broekema]. The UML models for system development are operational but the UML business extensions are currently still in development. Archimate will be fully developed at the beginning of 2005 but is not yet operational.

Criterion: For the creation and maintenance of models to be efficient, a tool must exist
ARIS, UML and IDEF are all operational and there are already tools that support these modelling languages. Both IDEF and UML are open standards and a large number of supporting tools exits. ARIS is only supported by the ARIS toolset as it is proprietary of IDS Scheer. Archimate is also an open standard and currently negotiations are taking place with the following tool vendors: Adaptive, BizzDesign, Popkin and ASG [Lankhorst].

3.9.3 Overall judgement

Our overall judgement is that IDEF and UML are currently too technical to be of use for enterprise modelling as stakeholders can also be of a non technical background. The UML business extensions are also still 'under construction' which currently makes them unusable. ARIS is currently the most usable enterprise modelling language, especially for business process modelling. It is likely that Archimate will become the standard for architectural modelling language in the years to come. Archimate is currently the only architectural modelling language available and because it is also backed up by a large consortium of organizations that combined have a fast influence on at least the Dutch market it could also become a standard.

3.10 Conclusion

We conclude that currently four enterprise modelling languages exist that can be used to model the enterprise architecture in principle and that are used throughout the industry. These enterprise modelling languages each belong to a specific layer of the EMF. UML and IDEF are both modelling languages that originate form modelling languages for system design and have evolved in such a way that they are partly suitable for business process modelling. IDEF is no longer evolving which will in time result in no longer meeting the requirements of the market. UML is currently still being developed to make it suitable for practical use in the area of business process modelling and it is therefore not yet usable. ARIS is a modelling language that focuses on business process modelling. Currently it is the most usable enterprise modelling language for business process modelling. Archimate is a modelling language that can be used for architectural modelling. Archimate is currently the only architectural modelling language but it has not yet been extensively used throughout the industry.

4 CHAPTHER FOUR: PRACTICE OF ENTERPRISE MODELLING

4.1 Introduction

Often, not only the type of modelling language or the quality of a modelling language but the architectural process, architectural maturity of an organization or a tool can be of influence for the use of an enterprise modelling language. We will discuss this subject throughout this chapter. Secondly we will introduce guidelines for modelling languages that enable architects to determine which type of enterprise modelling language should be used for a given situation. Finally we will demonstrate how modelling fits into the entire architectural process.

4.2 Tooling

4.2.1 Introduction

When constructing models it is highly recommendable to make use of a tool as it is time efficient. There are several tools for enterprise modelling available. In this section we will discuss the use of different types of tools and how a choice for a specific tool can influence the architectural process.

4.2.2 Tools for sketches

Tools which can be used to create sketches are used to create digital models of the paper sketches, so that they can be distributed or be presented to others. Tools that can be used for constructing sketches are usually not specifically designed for enterprise modelling. One of these objectives is the creation of models that consist of simple symbols. The most common of these tools used are Microsoft Word, Microsoft PowerPoint, Microsoft Visio and Adobe Acrobat.

4.2.3 Architectural modelling tools

Architectural tools are used to create models in the second layer of the EMF. During this research we have found several differences between the tools that can be used for creating architectural models.

Several architectural modelling tools are not based on any architectural modelling language and only offer a certain amount of symbols that are commonly used. Architects usually make use of these tools to construct architectural models based on an architectural framework such as DYA, Zachman or TOGAF. Another reason for the architects to make use of these types of tools is that there is a lack of architectural modelling languages as Archimate is currently the only initiative in the Netherlands.

A second selection of modelling tools does have some connection to an architectural modelling language. These tools use symbols of an architectural modelling language but no more than that. An example is the tool BizzDesign Architect from the company BizzDesign. It makes use of the Archimate symbols and is based on the Archimate meta model but has no further connection. According to [Hoppenbrouwers] Archimate

has partly been based on the BizzDesign tool. The next step would be to entirely integrate the Archimate modelling language into the BizzDesign tool. By doing so, the tool could be used to construct models that abide by the rules belonging to Archimate and in accordance with the Archimate architectural vision.

4.2.4 Business process modelling tools

There is a large number of business process modelling tools available. Most of them are based on some business process modelling language but the possibilities offered differ. Most tools offer the capability to construct models for a number of modelling methods. It is not unusual to encounter tools that can be used to create e.g. IDEF and UML models. The extent to which it is possible to construct models can also differ. As with the architectural modelling tools, these tools sometimes offer the capability to perform risk and impact analysis.

4.2.5 Web based models

Several tools offer the capability to generate websites that can be placed on the internet or organizational intranets. Other tools, like e.g. ARIS or BizzDesign, offer the capability to construct models that are based on the hyperlink principle which makes it possible to link from a model to another model. If we take the BizzDesign tool as an example it is possible to construct descriptions of the technical infrastructure which exists of several network servers. By making use of hyperlinks it is possible to select a server and link to e.g. a business process in which the particular server plays a role [Lac2004]. We feel that the hyperlink technology is very useful as it can deliver information on demand.

4.2.6 Risk & Impact analysis

We previously discussed the concept of Risk & Impact analysis that a tool can offer as support for an architect. In this section we will discuss this concept as we have several remarks on it.

Risk & impact analysis is an automated solution that helps architects identify the results of change in the architecture. Before an architect wishes to make a change in the architecture it is possible to simulate the alteration in a tool that supports risk & impact analysis. The architect can make a change in the models belonging to the architecture and the tool will predict the impact it will have on the entire architecture. It is also possible to let the tool determine risks that can occur due to a change.

[Goeij] has concluded that these tools should not be used until an organization has a high level of architectural maturity. We agree as we find that risk & impact analysis requires that an organization is mature because predictions must be created from a solid basis. If the architecture itself is immature chances are that the models of the architecture will also be of 'poor' quality which will lead to predictions that are incorrect. Even if the models are of good quality predictions will be unusable as the actual architecture will differ from the models. Furthermore it is important that a tool is selected especially for an organization. As with any other software it is important that the organization itself is mature before an automated solution is introduced to support the organization.

If analysis is performed on architectural models the chance that a fault will be detected will be next to none. When an architect constructs these models it is more likely that faults will be discovered by the architect. This is because an architect will still have an oversight of the entire enterprise as the degree of details is still acceptable. Another argument that we provide is that the lack of detail in the architectural models will also exclude the detection of faults beforehand. An example can be the removal of a large network server from the architecture. If the architect is acquainted with the architecture it is likely that the architect will be aware of the risks and impact in the organization. Risk & impact analysis will not offer any benefit in this situation.

Business process models contain far more details compared to the architectural models. With this we might seem to imply that finding faults in the enterprise architecture will be more likely, which is correct but there is a downside. As the quantity of details grow, so will the quantity of faults in the models. Models are constructed by humans and the more work that is performed on constructing the models, the more faults the models will obtain. Modelling faults will lead to the detection of enterprise architecture faults when performing consistency checks. A lot of these faults may of course not exist in the 'real' situation.

A final problem is that consistency checks and risk analysis requires the models to be constantly up to date which requires a lot of commitment and discipline from employees throughout the organization. For these types of analyses to be successful they require that employees constantly update the models. As updating models is not the 'core-business' of the average employee this form of discipline will be hard to obtain. The phrase *'the chain is as strong as the weakest link'* applies to this situation. In large organizations, like multi-nationals, it is likely that the required discipline will never be achieved. The larger organizations are the ones that could require such analysis as it can contribute to the organizational transparency; small organizations will have an insight into their architecture as the architect will know every detail of the architecture without making use of tools or documents. We can conclude that it is highly unlikely that these tools will offer success to organizations by risk and or impact analysis. However, at this moment, there is not enough data available yet to confirm our ideas as the tools are not yet commonly used.

4.3 Enterprise modelling & Architects

4.3.1 Introduction

There have been several proposals for types of architects. In this section we will attempt to make a valuable contribution to this discussion by defining what an architect does from a modelling point of view. Secondly we will seek the relation to the current definitions of the architects. First we will describe several architects that are commonly used throughout the industry.

4.3.2 The architect

The title enterprise architect is the most commonly used definition for the architect that is responsible for the entire enterprise architecture. The enterprise architects main goal is to define architectural principles for the entire enterprise [Rijsenbrij05b]. The title domain architect is another common definition used to address a specific type

of architect. To understand the role that such an architect plays we will study the definition of the term domain.

'A territory over which rule or control is exercised' [Heritage]

Following this definition combined with the term of architect we can conclude that the title domain architect refers to an architect who rules over some territory. The territory itself has not yet been defined but we can assume that it is at least a part of the entire architecture. The title domain architect is used quite often without making any reference to the type of domain itself. In the following two sections we will discuss the most common types of domain architects.

4.3.3 Domain defined by architecture

In chapter one we have discussed that architecture is divided into three separate domains: business, information and technical. In this case the domain is defined as a subsection of the enterprise architecture. If the domain architect is defined in this manner the domain architect would be the owner of part of the entire architecture. According to the DYA method the following domain architects then exist:

- Business architect
- Information architect
- Technical infrastructure architect

In this case the enterprise architect will define the architectural principles for the enterprise architecture. The domain architects will then make use of the enterprise architectural principles to define architectural principles for the business domain, the information domain and the technical infrastructure domain.

4.3.4 Domain defined by business architecture

Another common type of the domain architect is defining the domain according to the structure of the organization. Several territories are defined and an architect is assigned to one of these territories. As organizations differ we can not define any standard types of domain architects but here is an example:

- Organization divided into Divisions

This domain architect concentrates mainly on a demarcated part of the organization. The organization is decomposed and an architect is assigned to a part of this hierarchy. An example is [Dijk] who has been assigned as domain architect for the Rabobank division for 'Informatisering Betalen en Sparen'. In this role Dijk is responsible for defining architectural principles for the domain 'Informatisering Betalen en Sparen'. With the example we can conclude that this type of domain is responsible for a territory that covers some section of the business.

In [Rijsenbrij05b] the author has explored the definition of the domain architect and he has come to the conclusion that a domain architect must be defined around a specific business territory such as education, research or human resources. A criterion

for appointing a person as a domain architect is that the person must posses knowledge and experience of a specific domain. The author points out the advantage of the business being the leading influence for a specific domain and not technology. This principle indeed ensures that the technology is used to realize business demands instead of the business constantly having to adapt to 'useless' technology which eventually will 'slow' down the organization.

Rijsenbrij also discusses an application architect and a workplace architect. In contrary to the domain architect the application architect concentrates on the applications within the enterprise. The workplace architect focuses on internal roles (e.g. a HRM manager) and external roles (e.g. a client) that belong to an organization and is best compared to interior architects who are common in physical architecture. We must point out that the application architect and workplace architect can also be considered a form of a domain architect. The territories over which they rule are those of applications and workplace.

It is hard to define a general domain architect for all organizations. Different organizations will have different needs and it is best to define the territory of a domain architect according to these needs. A result of defining the territory for each specific organization is that the title domain architect can become ambiguous. The title application architect clearly refers to the domain applications as its territory. It is wise to make a reference to the territory a domain architect is responsible for as this will prevent confusion. We mentioned the example of [Dijk] and another example can be extracted from [Rijsenbrij05b]. If a domain architect is responsible for human resources the architect is best referred to as the domain architect for human resources.

4.3.5 Modelling role

We will now define the roles that architects can take on. We have made use of the enterprise modelling framework to differentiate the types of architectural roles. We have based the architectural role on the modelling work they perform and not by a domain defined some other way. We have distinguished the following roles and therefore architects:

- Strategic role
- Tactical role
- Operational role

These roles differ from the previous domain definitions but there is a strong relationship between them. The previous definitions for an architect are highly dependent on an architectural vision or on the organizational structure. Our architectural roles can co-exist within every architectural vision and organization as we will demonstrate.

Enterprise role
Defining the architectural principles for the enterprise architecture can be considered a form of strategic work for the organization. An enterprise architect can therefore be considered an architect that fulfils the strategic role.

Domain role
The domain architects once again perform strategic work. After the defined enterprise strategy is defined, the architectural principles for each specific domain needs to be defined thus the domain architect will also fulfil the strategic role. In contrary to the

enterprise architect the domain architect requires a tactic to achieve its strategy for the domain and therefore the domain architect will also fulfil the tactical role. It is also important that the architecture is actually implemented in the domain so an operational role is also required.

In this situation each architectural domain would require a strategic role, a tactical role and an operational role. It is possible that one domain architect will take on the three roles but the roles can also be divided over more then one person. This is a choice that must be made depending on the demands of a particular organization.

As we can see the three architectural roles can act independently from the domains defined throughout the organization. The advantage is that alignment between the various domains is achieved on a strategic, tactical and operational level because the constructed models are not bound by any domain.

4.4 Guidelines for Enterprise modelling

4.4.1 Why guidelines

By making use of the EMF we can offer various guidelines that can support the architects when he or she is creating an architecture. The guidelines have been defined in accordance to the attributes defined for each type of enterprise modelling in the second chapter. By making use of the guidelines an architect can discover which layer of the EMF he or she must target.

4.4.2 Guidelines

Guideline one: Level of use
An architect must make use of models out of the first layer of the EMF if the level of use is a strategic one, an architect must make use out of models of the second layer of the EMF if the level of use is of a tactical one or the architect must make use of models out of the third layer of the EMF if the level of use is of an operational level.

Guideline two: Architectural project group
If there is a team of architects the modelling work can be divided according to 'experience'. Architectural trainees, business process engineers and software engineers should make use of business process models as these people are inexperienced. Business process models can act as a guide to the inexperienced as they provide rules and guidelines for modelling. The more experienced architects can make use of sketches.

Guideline three: User group
Stakeholders can be roughly divided into three different groups. If the user group consists of higher executives an architect must make use of sketches. Project leaders, unit managers and other middle management people belong to the user group for which architectural models should be used. Business process models can be used for the specialists within an organization.

Guideline four: Architectural awareness
Architectural awareness can be achieved by making use of sketches as this is one of their objectives. If an organization is aware of architecture then architectural models

can be used. When there is high awareness of architecture an organization can make use of business process modelling to implement its architecture.

Guideline five: Quantity of details required
The degree into which a model is a complete reflection of some reality, depends on the quantity of details of which a model consist. The required completeness of the models depends on the organizational or stakeholder requirements. If much detail is required then business process models should be used, if fewer details are required then sketches are enough.

Guideline six: Architectural process
The three processes (strategic dialogue, architectural services and developing with(out) architecture) each make use of one specific layer of the enterprise modelling framework. Sketches have a strategic nature and can therefore be constructed in the strategic dialogue. Architectural models are constructed by architectural services as they define the architecture. The third process, developing with architecture, can create business process models as both are of an operational nature. Modelling as part of the entire architectural process is visualized in figure 4.1.

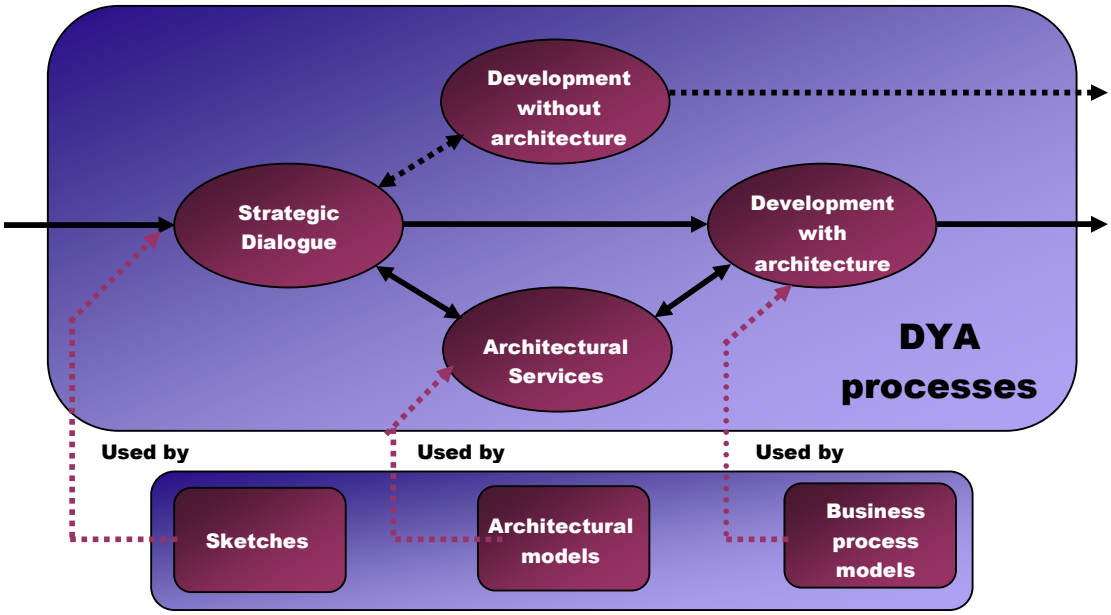

Figure 4.1: Modelling as part of the architectural process

4.5 Integration of the EMF layers

4.5.1 Introduction

The several enterprise modelling languages are divided over the three layers of the EMF. As each layer of the EMF has its own objectives the layers of course do have some relationship. As we have discussed the three layers of the EMF as individual layers there is also need for integration of the layers. In this paragraph we introduce several opportunities or possibilities for integration.

4.5.2 Current problem

At this moment the enterprise modelling languages are developed to fit into one of the three layers of the EMF. By concentrating only on one layer the problem arises that the enterprise modelling languages of the different layers don't connect to each other.

4.5.3 From Sketches to BPM and back

As we have discussed in chapter three, ARIS is currently the best choice for process modelling. Archimate is currently the only architectural modelling language and architectural models should be constructed with it. A problem that occurs is that there is no alignment between the two modelling languages which means that it is hard to construct ARIS business process models from Archimate architectural models. There is some similarity between ARIS and Archimate as process modelling is possible in both, ARIS does however offer a broader insight into the business processes.

Aligning sketches with Architectural models might seem difficult at first because the models are highly dependent on an architect. The contrary is in fact true as the architect can construct the sketches in such a way that they can easily be aligned with an architectural modelling language. It is more difficult to align the architectural models with the business process models.

Feedback from business operations to the executives is of importance for an organization. This principle can also be used when constructing models. Architects must construct their models in such way that they will 'connect' to each other by making use of feedback. Because sketches are used as basis for the creation of architectural models it is important that an architect who is creating the architectural models informs the architects who created the sketches about 'issues'. An issue can e.g. be a flaw in the sketches detected by an architect constructing the architectural models. If such 'issues' are clearly communicated the sketches and architectural models will 'connect' better to each other. The same principle can be used between the creators of the architectural models and the creators of the business process models. We have visualized this in figure 4.2.

Of course this will not be possible for all the models and an architect should not try to 'force' the models to 'connect' as this would lead to modelling becoming the objective instead of modelling assisting to achieve the objective.

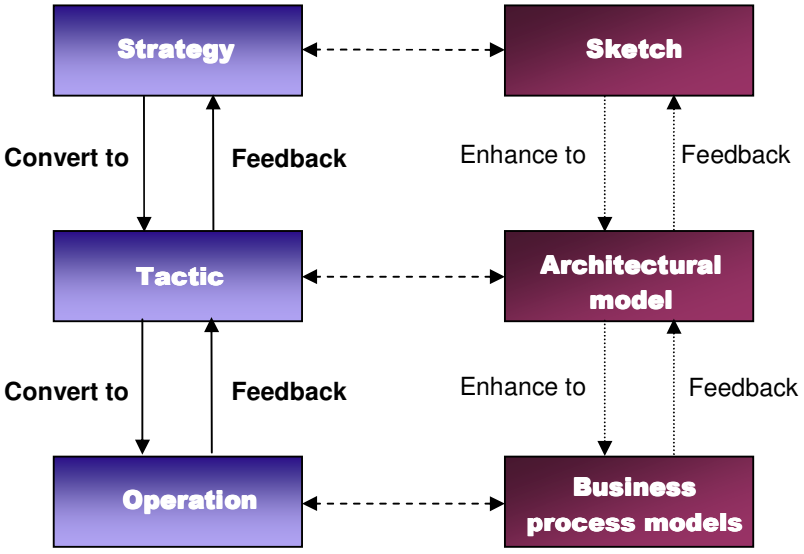

Figure 4.2: Modelling cycle

4.6 Conclusion

We conclude that tools can be of use when constructing enterprise models but one must be aware that tools can differ. Further we have determined that the use of risk & impact analysis will not automatically lead to business advantages.

For an architect to determine which type of models need to be constructed we have introduced six guidelines. These guidelines will help the architect to determine which models are required. The guidelines are based on the following factors: level of use, architectural project group, user group, architectural awareness, quantity of details required and the architectural process. Furthermore we have determined that together with the three types of enterprise modelling three types of architectural roles exists. The strategic role can make use of sketches during the strategic dialogue. The tactical role will determine the organizational tactic by transforming the strategy into a tactic as part of the architectural services. The operational role will transform the tactic into concrete projects that will develop under architecture.

5 CHAPTHER FIVE: CONCLUSION

5.1 Research questions

5.1.1 Introduction

As described in chapter one our research has been conducted around the main research question which was divided into several sub questions. In the previous chapters we have provided our findings and will now apply them to answer our research questions starting with the sub questions.

5.1.2 Sub question one

Our first sub question was defined as follows:

Which modelling languages are currently available?

We have determined that the unified modelling language (UML), the architecture of integrated system (ARIS) and the integrated computer aided manufacturing definition languages (IDEF) are currently available and applied enterprise modelling languages throughout the industry. Further we have determined that Archimate is a new enterprise modelling language that is about to be applied.

5.1.3 Sub question two

Our second sub question was introduced to determine the properties of each enterprise modelling language so that it is possible to distinguish them. The sub question is as follows:

Of which modelling methods do the modelling languages consist?

ARIS and UML are modelling languages that consist of several separate views and one combining overview. The languages consist of a number of modelling methods that each belong to one of the views. ARIS consists of modelling methods to describe the organizational strategy & objectives, value chain, organization functions, organizational structure, organizational data objects and the business processes. UML, together with the business extensions, consist of modelling methods to describe the organizational objectives, organization structure, the relations between actors and business functions, organizational data objects, the business processes and the application and infrastructure.

IDEF is a modelling language for which no framework and or views have been defined. IDEF is a set of modelling methods that are used to describe the organizational structure, organizational data objects, organizational functions and the business processes.

Archimate is a modelling language which focuses on the business architecture, information architecture and technology architecture. Contrary to the current enterprise modelling languages Archimate revolves around viewpoints and not around modelling methods. The viewpoints can be dynamic which results in dynamic models.

5.1.4 Sub question three

Our third sub question was introduced to determine the objectives of each enterprise modelling language so that it is possible to offer a basis for comparison. The sub question is as follows:

What are the main objectives of the modelling languages?

To describe the objectives of the modelling languages we will make use of the enterprise modelling framework which distinguishes three types of enterprise modelling languages: sketches, architectural models and business process models. UML and IDEF have as objectives system engineering and business process modelling. ARIS primarily has as objective business process modelling and Archimate has architectural modelling as its objective.

5.1.5 Sub question four

Our fourth question was introduced to determine the quality factors for enterprise modelling languages. With the quality factors we are able to judge the quality of enterprise modelling languages. The sub question has been defined as follows:

What are the main quality factors of modelling languages?

As we have described the quality factors in detail throughout chapter three we will now only briefly mention them. They are as follows:

- Enterprise modelling languages must recognize the existence of stakeholders and recognize their importance to architecture.
- The resulting views of an enterprise modelling language must foresee in the need of the stakeholders.
- Stakeholders must be able to understand and interpret the descriptions offered by an enterprise modelling language.
- Enterprise modelling languages must adhere to some kind of standard.
- Enterprise modelling languages must posses transferable models.
- Enterprise modelling language must offer the capability to model the environment in which the enterprise functions.
- An enterprise modelling language must be adaptable.
- Enterprise modelling language must be able to model the complete enterprise.
- Enterprise modelling languages must be able to model the alignment between the architectural domains.

- An enterprise modelling language must be marked as fully developed by its creators.

- For the creation and maintenance of models to be efficient, a tool must exist.

5.1.6 Sub question five

Our fifth sub question was introduced to determine the level of experience with the enterprise modelling languages. This question was defined to separate the modelling languages which have proved themselves throughout the market. The sub question is as follows:

Is there any experience with the modelling languages?

As IDEF and UML have been around for a while and both have become a standard a lot of experience is available. We found that IDEF was primarily used throughout the United Stated by large corporations such as Lockheed Martin, General Motors and governmental organizations such as the US Air Force. Furthermore, [Schekkerman] mentioned the use of IDEF for a project at the Royal Dutch Army. UML has become a world wide standard that is practiced by an enormous amount of organizations. Both UML and IDEF are especially used for systems engineering.

ARIS is currently the number one toolset for business modelling. Organizations that make use of ARIS are e.g. British Telecom[IDS04] and KPN[Claus]. Archimate is not yet fully operational but a number of pilots have been performed at e.g. the Dutch tax office [Belastingdienst03].

5.1.7 Sub question six

Our sixth sub question was introduced to determine the influences on which a choice for an enterprise modelling language is based. Our objective was to translate these influences into guidelines. The sub question is as follows:

What are the influences that affect the choice for a modelling language?

As we have discussed the influences and guidelines in detail throughout chapter four we will now only briefly mention them, they are as follows:

- Level of use
- Architectural project group
- User group
- Architectural awareness
- Quantity of details required
- Architectural process

5.1.8 Sub question seven

Our seventh sub question was introduced to determine if any enterprise modelling languages could be combined. The sub question defined was as follows:

Is it possible to combine enterprise modelling languages?

We can conclude that enterprise modelling languages should not be combined if they belong to the same layer of the enterprise modelling framework. Enterprise modelling languages that belong to the same EMF layer will have similar objectives but will differ in the manner in which these objectives are achieved. It would be unwise to combine modelling languages as the several modelling methods of which they consist will not be compatible. A second problem that can occur is that work is done double to satisfy the modelling rules of each modelling language.

It is desirable to combine modelling languages that belong to different layers of the EMF as their objectives differ and they can be an extension to each other.

5.1.9 Sub question eight

Our eighth and final sub question was introduced to determine the consequences that occur when enterprise modelling languages are used. Our objective was to determine the benefits and drawbacks of the use of modelling languages. The sub question was as follows:

What are the consequences of making use of an enterprise modelling language?

In the second chapter we have discussed the benefits of the use of enterprise modelling languages. Further we discussed throughout chapters two, three and four modelling can also have disadvantages. We have discovered that the main disadvantage of modelling is that modelling can replace the original objective, gaining or finding new business advantages, and that constructing models will become the objective. To avoid this issue one should keep in mind that modelling can be used as support for the entire architectural process.

5.1.10 The main research question

We have performed our research and have provided answers to the sub research questions. With our findings we are now able to provide an answer for our main research question. The main research question is as follows:

How can an architect create a usable description of an architecture?

An architect can create a usable description of an enterprise architecture by making use of enterprise modelling languages. Enterprise modelling languages will contribute to the transparency of the enterprise and therefore help organizations be adaptive. Enterprise modelling should be performed on three distinctive levels. The overall strategy should be defined by making use of sketches in the strategic dialogue. The strategy can be translated into a tactic by making use of architectural models as part of the architectural services. Transforming the tactic into an operational environment

can be done by developing under architecture and making use of business process models. All three types of enterprise modelling must be performed as successful enterprise architecture will depend on the feedback between the overall business strategy and the actual business operations. This is supported by [Forrester04-1] which states that business process modelling forms the core of the enterprise architecture.

When creating architectural models an architect should make use of the Archimate modelling language. When creating business process models an architect should make use of the ARIS modelling language.

5.2 Modelling in the coming years

We believe that this research has shown that organizations can benefit from the use of enterprise modelling. At several conferences and on other occasions we determined that enterprise modelling could become a 'hype' in the coming years. Several tool vendors are currently expanding rapidly. Further we determined that organizations are becoming very interested in the concept of enterprise modelling. Enterprise modelling is also described as one of the trends for 2005[Forrester04-2]. As with every hype enterprise modelling is surrounded by a lot of 'fairytales'. With this research we believe that Sogeti Nederland B.V and the Radboud University have delivered a valuable contribution to the do's and don'ts of enterprise modelling before the hype has maximized.

5.3 Research proposals

5.3.1 Aligning the EMF layers

As we have described there are three distinctive types of enterprise modelling. We have furthermore determined that modelling languages mainly focuses on one of the layers of the EMF. Currently it would be the responsibility of the architect to construct the models in such a way that the models of the different layers of the EMF will 'connect' to each other. Our first research proposal is to determine if it is possible to gain a higher degree of alignment between the layers of the EMF. If alignment is possible a second research proposal would be to extend the current or the development of new, enterprise modelling languages so that they are able to cover the third and second layer of the EMF.

5.3.2 Impact and risk analysis

At the moment we are sceptical of the concept of impact and risk analysis. The required organizational discipline will most likely not be achieved throughout large organizations and as they are the organizations that should benefit from these types of analysis we feel there is no need for risk and impact analysis. We have however determined that a 'hype' is forming around the concept of risk and impact analyses. At [Lac2004] we found that risk and impact analysis stood high on the agenda of several organizations. At this moment we have no scientific data to support our conclusion and we therefore propose that research be conducted into the area of risk and impact analyses.

APPENDIX A. RESEARCH PROCESS

A.1 Introduction

We started our research preparation by writing a research plan. Our research plan described the main and sub research questions, research domain, the research methodology and an approach to the research. We have visualized our research process in figure A1.1.

A2. Research question & domain

Our research domain and question have been defined before the actual research initiated. In the preparation phase two meetings had been organised with the constituent Sogeti Nederland B.V where the research was discussed and the research question and domain was determined.

A3. Research methodology

Our research kicked off with some informal interviews in which information was gained on how to commence the research. To gather data for our research we have also performed a literature study and several interviews. We conducted our literature study by searching for and analysing the information. All interviews have been conducted in a similar way. The interview questions have been determined in advance as was the interview goal. When an interview initiated the objectives of the interview were discussed with the interviewee.

Interviews have been conducted with the following people and organizations.

- Adaptive; Marc de Goeij
- Telematica Instituut/ Archimate, Marc Lankhorst
- CapGemini; Jaap Schekkerman, Hans Goedvolk, Frits Broekema
- KPN; Monique Claus
- Rabobank; Rogier Dijk
- Radboud Universiteit Nijmegen, Stijn Hoppenbrouwers, Erik Proper
- Sogeti; Martin van den Berg, Jan Hoogervorst, Ivo van Ouwerkerk, Leo Aarssen, Stefan Langerveld

Further information was provided by Jan Campschroer of Ordina and gathered at several conferences.

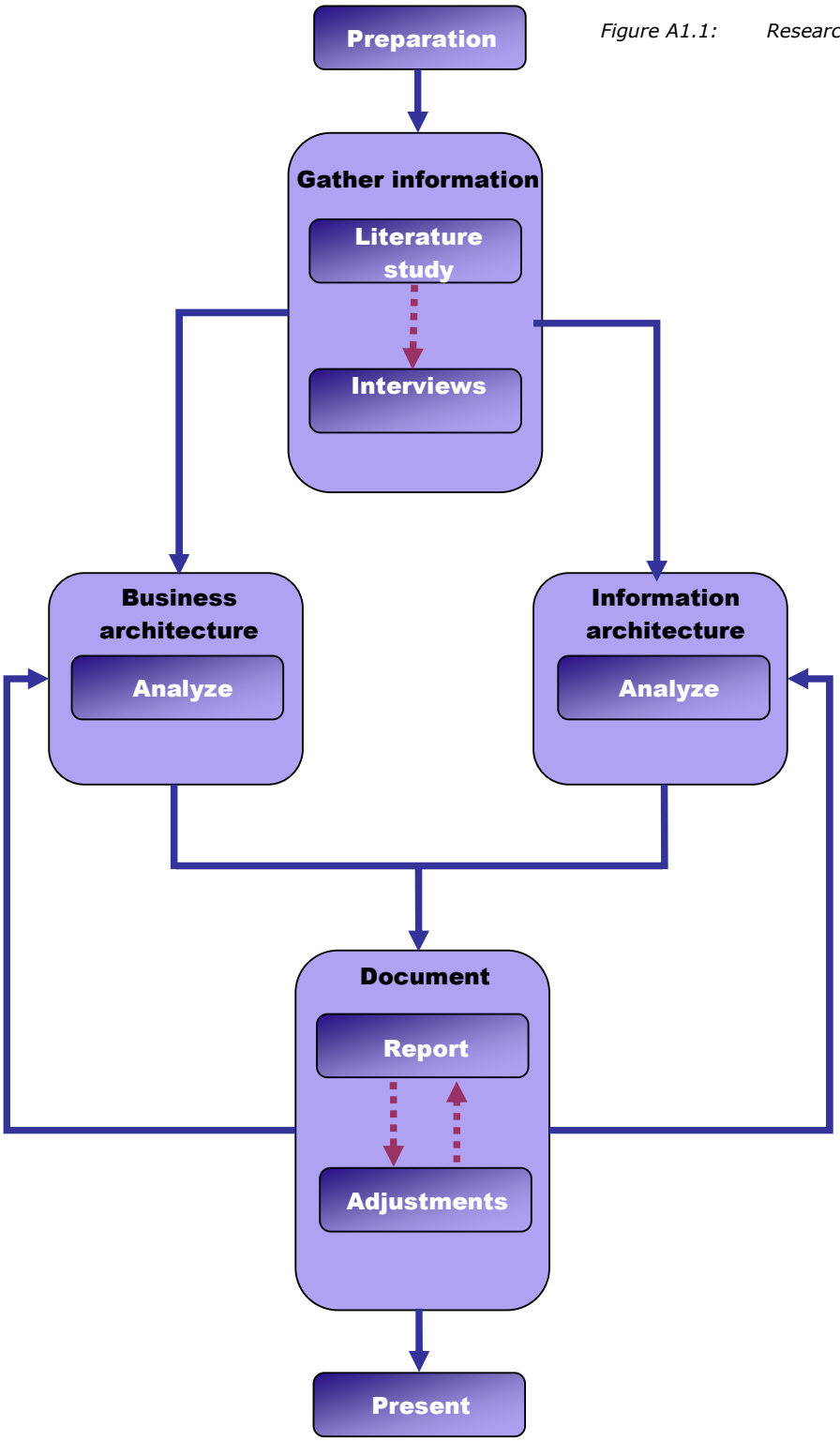

Figure A1.1: Research process

APPENDIX B. REFLECTION

B.1 Introduction

In this section we will glance back on the research period and determine successes and 'issues' that can be improved together with a reflection on what has been learned during the research period.

B2. Research preparation & Initiation

During the research preparation and initiation we have found that defining a concrete and clear research problem and question is difficult but critical. A clear insight into the research area and defining the problem require time and feedback from the constituent. Looking back at this phase of the research we can conclude it has been a success. Firstly, we feel that the research questions and deliverables defined in advance have been correct and have lead to results that meet the constituent's demands. Secondly, the constituent, in the person of Marlies van Steenbergen and Martin van den Berg, has described the performance of the main researcher, Christopher Magee as independent. With this the constituent means that a minimum effort in the form of guidance was needed. It is unlikely that this degree of independency and the research results would have been achieved if the preparation and initiations had been incomplete or incorrect. The constituent pointed out to the researcher that feedback from the researcher to the constituent is important as it enables the constituent to track the progress of the research. Even though the researcher has met the demands of the constituent it was sometimes hard for the constituent to track the progress of the research being conducted. We can therefore conclude that the preparation & initiation phase have been a success but that during future research feedback between constituent and researcher, and vice versa, should be more frequent.

B3. Research execution

We can conclude that the overall execution of the research and the results meet the predefined research plan and the expectations of the constituent. If we make a more detailed reflection we can say that the main researcher, Christopher Magee, has found the literature study a challenging part of the research. We found that the stream of information can be constant, enormous and highly interesting but in many occasions unusable. We mainly contribute this to the enthusiasm of people that were in some form related to our research especially in and around the Sogeti Nederland b.v. offices. During lunches and other informal occasions our research was often discussed in a brief manner. In a matter of time Sogeti employees started walking in and out of the main researcher's office with information in the form of numerous books, articles and names of individuals who could be interesting for our research. We can conclude that when discussing our research we should offer others a more detailed description of the research domain to control the constant unsolicited, but highly appreciated, stream of information. If we look at the information provided by the constituents it was beneficial as they were aware of the research in more detail.

B4. Research risk

A research risk encountered during the research is that of illness. We can say that the impact of such a small problem is larger that the main researcher had anticipated in advance. After being ill the researcher found it hard to 'get back on track'. We found that activities are clustered and depended on each other in such a way that delay can

cause a chain of unwanted occurrences. We therefore conclude that in the future research risks must be more carefully analyzed.

B5. Machine vs. Man

Over the period of this research the opinion of the main researcher on the relationship between man and machine has changed. Before the research the main researcher found that both the machine and mankind are very similar. We can now say that a difference has been established.

If we look at the business, mainly a collection of humans, and at the technology, mainly a collection of machines, we determined that there is a difference. The technology seems to be easier to model, thus easier to understand, compared to the business. Behaviour of the technology is easy predictable, as mankind has programmed it to display some sort of behaviour it will do so exactly. The behaviour of the business seems to be somewhat unpredictable. We can 'program' the business to perform some sort of behaviour through rules, procedures, predefined processes and models. Even though the behaviour has been 'programmed' in advance the business will never behave as programmed as humans seem to ignore it. So, the unpredictable business is a result of unpredictable humans.

The predictability of the result of this behaviour seems to be just the opposite. A collection of humans will be able to gain a set objective even if they ignore their predefined behaviour or if the objective itself is partly unclear. A machine on the other hand will only result in the predefined objective if its objective has been perfectly described which it hardly ever is. Our conclusion: the result of unpredictable human behaviour is predictable; the result of predictable machine behaviour is unpredictable. A collection of humans seems to be self-regulating and a collection of machines are unable to regulate themselves.

The alignment between business and technology is therefore a challenge. The business will offer incorrect input into the technology, as it in practice often does, which will in turn create incorrect and unusable output for the business. Even though the business will be self-regulating and will produce the expected results one can stop and ask the question: *'why are we spending so much money on technology if it does not provide what we ask for?'*. Of course it is exactly providing what the business asked for but it does not provide what the business wanted. In fact, in the coming years this problem will remain unsolved.

All this has its influences on the concept of enterprise modelling as it covers both a collection of machines and a collection of humans. With this we have not only advised but also learned that modelling should be used as a tool and not as the solution.

B6. Main researcher's impression

The impression of the main researcher is that the research has been successful in more than one way. We feel that we have successfully fore filled the research and have identified several interesting concepts. We also feel that the cooperation between the Radboud University and Sogeti Nederland b.v. has been successful on several levels. Both parties have benefited from the research and the main researcher has found it to be an enjoyable and educational time being posted as an 'external' at Sogeti Nederland B.V.

We are also grateful for the guidance of Sogeti Nederland B.V which was of quality. We also appreciate the high degree of freedom that the main researcher enjoyed during the research. We especially appreciate the support of Marlies van Steenbergen and Martin van den Berg as they made a lot of effort to guide the main researcher during the research, 'aligning' the main researcher with the provided research and 'integrating' the main research into the Sogeti organization.

We can say that the interviews were surprisingly present. All interviewees have been highly enthusiastic, cooperative and have strived to make the interviews possible. We have not only enjoyed the interviews but also the discussion in which all interviews resulted.

Furthermore the information, discussions and guidance offered by the second constituent Daan Rijsenbrij is greatly appreciated. We appreciate the effort of the second constituent to keep the researcher up to date on current issues and to constantly challenge the main researcher to reflect on these issues.

TERMS

Actor:
An element of the Unified Modelling Language. An actor represents a coherent set of roles played by a user of a system being modelled.

Adaptive Enterprise:
An Adaptive Enterprise is one that can quickly respond to and capitalize on change for business advantage. It is the ultimate state of fitness: business and IT perfectly synchronized

Alignment:
The arrangement of the architectural domains of the enterprise to support the overall purpose of the enterprise

Business function:
An activity or a group of activities which support one aspect of furthering the mission of the organization describing what is done within the organization independently from the organization structure.

Business object:
Any object that models the real world which is of relevance to the enterprise. Common examples are Client, Supplier and Employee.

CRM:
Customer Relationship Management. A term for software solutions that help enterprise's manage customer relationships

Data modelling:
Data modelling involves considering how to represent data objects within a system, both logically and physically.

Decomposition:
The separating or analyzing an element into component units.

Digital Architecture:
A consistent set of rules and models that guides the design and implementation of processes, organizational structures, information flows, and technical infrastructure within an organization.

EMF:
Enterprise Modelling Framework, a visualization of enterprise modelling.

Enterprise:
An economic organization or a non profit organization.

Enterprise model:
An enterprise model is an abstract description of the behaviour belonging to an existing or required enterprise, simplified by ignoring certain detail.

Enterprise modelling language:
An enterprise modelling language is a collection of methods with the objective of creating enterprise models.

ERP:
Enterprise Resource Planning. A strategy that integrates the enterprise functions such as manufacturing, finance and distribution to balance and optimize the enterprise's resources. ERP is usually supported by ERP software like SAP, BAAN, ORACLE and JDedwards.

Framework: A logical structure for classifying and organizing information.

IAF:
Integrated Architecture Framework. Architectural approach developed by GapGemini.

Model:
A model is an abstract description of behaviour belonging to a system, simplified by ignoring certain detail.

Operation:
Any action resulting in a change of the organization as described in the tactic.

Process flow:
A logical and sequential series of activities needed to complete a business function.

Service Oriented Architecture: One speaks of service oriented architecture if the separate layers of architecture deliver service to one and other.

Strategy:
A general direction to achieve a desired state in the future.

Swim lanes:
A partition on diagrams for organizing responsibilities for actions.

Tactic:
A detailed plan or course of action by which describes how a strategy is to be implemented.

TOGAF:
The Open Group Architecture Framework. An open standard for architecture.

LIST OF FIGURES

REFERENCES

[Aarssen] Leo Aarsen, *"Interview with Leo Aarssen "*, Architect for Sogeti Nederland B.V. 2004

[Agile alliance] Agile Alliance *, Business Modeling with UML: A Business Process Centered Architecture*, Agile Alliance 2003

[Archimate04] van Buren et Al, *"Mapping between Archimate and standards"*, Archimate project deliverable D2.2.23B 2004

[Ashby] Rosh Ashby, *"An introduction to cybernetics"*, Chapman and Hall LTD London 1957

[Berg] Martin van den Berg*, "Interview with Martin van den Berg"*, Service line manager and thought leader for Sogeti Nederland B.V. 2004

[Belastingdienst03] Belastingdienst, *"Presentatie ICT kenniscongres",* Centrum voor proces-en productontwikkeling belastingdienst 2003

[Broekema] Frits Broekema, *"Interview with Frits Broekema",* Architect for CapGemini 2004

[Burgt04] Therry van den Burgt*, "Onderzoek naar architectuur tools"*, Sogeti 2004

[Claus] Monique Claus, *"Interview with Monique Claus",* Architect for KPN 2004

[Dijk] Rogier Dijk, *"Interview with Rogier Dijk"*, Information manager for Rabobank 2004,

[DYA01] Luijpers et al, *"DYA:Snelheid en samenhang in business- en ICT-architectuur"*, Tutein Nolthenius 2002

[DYA04] Luijpers et al, *"Dynamic architecture: How to make it work"*, John Wiley & Sons Inc. 2005

[DYA05] Marlies van Steenbergen and Martin van den Berg, *"DYA: Stap voor stap naar professionele enterprise-architectuur"*, Ten Hagen Stam uitgevers 2004

[Errison-Penker] Hans-Erik Eriksson, *Magnus Penker, Business modeling with UML: business patterns at work*,John Wiley & Sons, Inc 2000

[Forrester04-1] Henry Peyret et al, *"Trends 2005: Business process modelling and enterprise architecture tools"*, Forrester research ,Inc. 2004

[Forrester04-2] Henry Peyret et al, "*Getting value from enterprise architecture tools"*, Forrester research ,Inc. 2004

[Gartner04] Bill Rosser, *"Presentation Gartner Symposium ITxpo"*, Gartner, Inc. 2004 Cannes

[Giga03] Gene Legenza, *"Discussing enterprise architecture with business executives"*, Giga Information Group 2003

[Goedvolk] Hans Goedvolk, *"Interview with Hans Goedvolk"* Architect for CapGemini 2004

[Goeij] Marc de Goeij*, "Interview with Marc de Goeij"*, founder and account manager for Adaptive 2004

[Heritage] Heritage, "*The American Heritage® Dictionary of the English Language",* Fourth Edition 2000 by Houghton Mifflin Company.

[Hoogervorst] Jan Hoogervorst, *"Orientation interview with Jan Hoogervorst",* Management Consultant Sogeti 2004

[Hoppenbrouwers] Stijn Hoppenbrouwers, *"Interview(s) with Stijn Hoppenbrouwers"*, Researcher for the Radboud University of Nijmegen and member of the Archimate project team 2004

[Hp03-2] Hewlett-Packard, "*The HP IT Service Management s(ITSM) Reference Model"*, Hewlett-Packard Development Company 2003

[HP03] Hewlett-Packard, *"Hp Federal Brochure"*, Hewlett Packard Development Company 2003

[IDEF] IDEF, *"IDEF website figures"*, IDEF website can be fount at http://www.IDEF.com

[IDS] IDS, "*ARIS BSC: Rapid prototyping for strategic performance management",* IDS scheer

[IDS04] IDS Scheer, *"ARIS in British Telecommunications",* IDS Scheer UK 2004

[ING04] Marc Lankhorst, *"Presentation ING bank case"*, Telematica instituut 2004

[ITIL-i] The Office of Government Commerce, *"ITIL: IT infrastructure library"*, The office of government Commerce can be found at http://www.ogc.gov.uk

[KPN04] van Buitenen et al, *"Business technologie: visie en implementatie"*, Koninklijke KPN N.V 2004

[Lac2004] Landelijk architectuur congres, *"speakercorner BizzDesign and Pokin Software",* Landelijk architectuur congres 2004

[Lankhorst] Marc Lankhorst, *"Interview with Marc Lankhorst"*, Telematica institute and project leader for Archimate 2004

[Lankhorst04] Marc Lankhorst et at, *"Archimate Language Primer"*, Telematica institute 2004

[Lankhorst05] Mark lankhorst et al, *"Enterprise Architecture at Work : Modelling, Communication and Analysis",* Telematica institute, Springer 2005

[NIST93] National Institute of Standards, information can be found at the following site *http://www.nist.gov*

[OMG] Object Management Group, more information can be found at the following site *http://www.omg.nl*

[Ouwerkerk] Ivo Ouwerkerk, *"Interview Ivo ouwerkerk",* Architect for Sogeti Nederland B.V. 2004

[Oxford] Edmund S. Weiner (Editor), John Simpson (Editor), "*The Compact Oxford English Dictionary"*, New edition by Oxford University press.

[Pan-Wei] Pan-Wei, *Effective Business Modelling with UML: Describing Business Use Cases and Realizations*, Rational Rose Singapore 2002

[Proper] Erik Proper, "Interview with Erik Proper", Professor for the Radboud University 2004

[Reijers, Aalst03] Hajo Reijers, Wil van der Aalst, *Formele methoden in business proces management*, Technical University Eindhoven in Informatie 2003

[Rijsenbrij04] Rijsenbrij et al*, "Architectuur, besturingsinstrument voor adaptieve organisaties"*, Lemma 2004

[Rijsenbrij05] Daan Rijsenbrij*, "Collegedictaat Digitale Architectuur",* Radboud University 2005

[Rijsenbrij05b] Daan Rijsenbrij, "Architect in de digitale wereld: vechten tegen complexiteit", TIEM nr. 7, March-April 2005

[Rijsenbrij-i] Daan Rijsenbrij, "*Website Daan Rijsenbrij"*, http://home.hetnet.nl/~daanrijsenbrij/vision/eng/begrip1.htm

[Schekkerman] Jaap Schekkerman, *"Interview with Jaap Schekkerman"*, Architect for GapGemini and founder of the Institute for Enterprise Architecture Developments

[Schekkerman04] Jaap Schekkerman, *"How to survive in the Jungle of Enterprise Architecture Frameworks: Creating or choosing an Enterprise Architecture Framework"*, Tratford Publishing 2004

[Spiekerman03] Sarah Spiekerman, *"Process and organization modelling",* Institute of Information Systems Humboldt University Berlin 2003.

[Van Vliet05] Mario van Vliet, *"College Systeemontwikkeling les BPR"*, Radboud University 2004

[Vermeulen, Aaldijk] Erik Vermeulen and Rob Aaldijk, *"Modelleren van organisaties"*, Landelijk Architectuur Congres 2001

[Vernadat] F.B. Vernadat*, "Enterprise modelling: Objectives, constructs and ontologies"*, University of Metz and Eurostat European Commission 2004

[Wikpedia] Wikpedia Foundation, *"Wikpedia free encyclopedia"*, Internet: http://en.wikipedia.org/