MASTER

Enterprise architecture

the selection process of an Enterprise Architecture Toolset to support understanding and governing the enterprise

Dragstra, Paul

Award date:
2005

PHILIPS

TECHNISCHE UNIVERSITEIT EINDHOVEN
Department of Mathematics and Computing Science

MASTER'S THESIS

Enterprise Architecture

The selection process of an Enterprise Architecture Toolset to support understanding and governing the enterprise

By
P. Dragstra

Institute:	Technische Universiteit Eindhoven
Supervisor:	Prof. Dr. K.M. van Hee and Ir. W.F. Rietveld
Period:	March 2004 – April 2005
Company:	Royal Philips Electronics of the Netherlands
Department:	Philips International B.V. Corporate IT
Coach:	Ir. C. Wielinga and Mr. F. Butstraen

A. Preface

This thesis is the result of the graduation period from the study 'Computer Science and Engineering' at the Technische Universiteit Eindhoven (TU/ e). The graduation took place in the field of Information Systems, with Enterprise Architecture as the area of expertise, and was carried out at Philips.

Philips Electronics of the Netherlands is a global leader in colour television sets, lighting, electric shavers, medical diagnostic imaging and patient monitoring, and one-chip TV products. Its 164,000 employees in more than 60 countries are active in the areas of lighting, consumer electronics, domestic appliances, semiconductors and medical systems. The graduation took place at the department Corporate IT.

Philips wants to move 'towards one Philips'. That means that the different areas have to be able to cooperate. That can result in sharing the same resources in the field of IT. To find out which services are suitable to become a shared service and to adjust the IT services to the different businesses the use of architectures is necessary.

In order to talk about architecture a formalism is necessary. An Enterprise Architecture tool provides such formalism and it supports the architecture methodical. The tool makes a dialogue between the business and IT possible and it is also capable of showing future changes in the architecture.

Finally, I would like to thank my supervisors Erica Rietveld, Kees van Hee, Kees Wielinga and Frank Butstraen for their ideas, comments and support. Besides them I want to thank Maarten Leurs for his support and work in this field. In particular I would like to thank Jan, Marianne, Mark and the rest of my friends and family for their love and support during this graduation process and throughout my study.

Paul Dragstra,
Eindhoven, April 2005.

B. Abstract

"The improvement of understanding is for two ends: first, our own increase of knowledge; secondly, to enable us to deliver that knowledge to others." [John Locke (1632-1704), British Philosopher]

Understanding is an important issue in enterprise governance. Decision-making in the enterprise can be improved by a better understanding of the structure, processes, and functions of the enterprise or by understanding the impact of the decisions. By the introduction of information systems, enterprises are becoming more and more complex to understand. Also customers are becoming more and more demanding and this results in the fact that understanding the enterprise becomes more necessary than ever before.

Enterprise Architecture is a possible solution for the lack of understanding. It is still a young profession and at this time there is limited consensus on definitions and methodology. Here it will be considered as a meta-architecture, thus it is a set of architectures. With Enterprise Architecture the different views and architectural descriptions can be mutual related. In this thesis these relationships are considered as the added value of an Enterprise Architecture. The definition of Enterprise Architecture, which will be used during this thesis, is based on the IEEE definition for architecture. That definition considers systems and systems can be split up in information systems, enterprise systems, aspect systems, and subsystems. The enterprise itself can also be considered as a system.

In this thesis, Enterprise Architecture is split up into three variants, mentioned as the three approaches. These approaches are:

- **IT-centric:** the most important objective is Business - IT alignment;
- **Business process-centric:** the most important objective is to manage and improve business processes;
- **Governance-centric:** the most important objective is to understand the business better and to govern it better.

Besides these main objectives of the three approaches, Enterprise Architecture can benefit the enterprise in another number of ways. For instance, these three approaches all result in better communication with the stakeholders.

Enterprise Architecture is a complex subject and the Enterprise Architecture frameworks are a methodology to support the architect in adopting an Enterprise Architecture. Frameworks provide a logic structure for classifying and organizing the different architectural descriptions of the enterprise. The legalisation in the USA, with the Clinger-Cohen Act and the Sarbanes-Oxley Act, has provided for a huge increase in the use of Enterprise Architecture and its frameworks. In this thesis two frameworks will be discussed. First the Zachman framework will be described because this was the first framework in the world and it is still important nowadays. The second framework is the Integrated Architecture Framework. This framework will be used in the situation of Philips.

Enterprise Architecture tools are developed to support Enterprise Architecture and frameworks in a formal and systematic way. Enterprise Architecture can be supported by all

kinds of tools. These tools consist of a modelling aid and a repository to store the information systematically. There are three main approaches in these tools. These tools are moving closer and closer to each other, so in the future there might become one kind of tool that fulfils in all the functionalities of these three kinds of tools. These approaches are:

- **EA top-down approach:** this approach is started at the top (strategies and drivers) and go down to the bottom (technical implementation). This is the executive's view;
- **EA bottom-up approach:** this approach starts at the bottom (with the technical implementation) and goes to the top (the strategies and drivers). This is the employee's view;
- **EA change management approach:** this approach starts in the middle to support some projects. The tools are only used for some projects. This is from the manager's or employee's perspective.

The second part of this thesis will be describing the tool selection process for Philips. Enterprise Architecture has to support Philips in their 'towards one Philips' strategy. This strategy has the objective in more cooperation in the field of architecture between the different divisions of Philips. Therefore Philips needs a tool to manage their IT both on corporate level as on divisional level. The main purpose is a better Business - IT alignment and to be able to make better decisions in the field of IT. Therefore the tool that suits the requirements of Philips best had to be selected.

The first step in the tool selection process was to generate a long list. This long list contained the main tools that were available and the three different approaches were taken into account. Therefore at least one tool from every approach was selected. During this selection process a Request For Information was sent to eight tool vendors. With the help of an RFI the long list was reduced to a short list of three tools. During this phase the main focus was on creating different models and relationships between the models. The models, which have to be generated and stored with the help of a tool, have to be in harmony with the IAF framework. After the RFI process, the three tool vendors from the short list were invited to give a presentation and demonstration of their tool. Based on these presentations and also based on the results to the RFI Metis from Computas was considered as the most promising tool for Philips. Therefore Metis was selected for a test case.

The main purpose of the test was to find out if the tool is able to check consistency, links, and overlap between different views of the same architecture. These aspects were checked by using the Hydra and the PHERA document during this test case. Therefore the As-is reference architectures of Hydra and PHERA were implemented in the Enterprise Architecture tool. Also a third view was created. This is the SITAR view and from the SITAR document things like strategies, business principles and IT principles were taken. After implementing these three views a fourth view was created. This view is based on the IAF framework and the input was generated from the PHERA, Hydra, and SITAR views. During the test case Metis did not show any imperfections.

The introduction of an Enterprise Architecture for Philips requires still a lot of work to do. First of all a better test case with Metis has to be held. During this test case a team server has to be used, to see how cooperation with Metis works. Second there has to be generated a maturity and willingness among the employees to use an Enterprise Architecture.

Contents

1 Introduction

A lot of IT projects could be better adjusted to the enterprise with the help of Enterprise Architecture (EA). In 2003 Gartner Inc. states that, *"IT project failures in industry and Government account for an astounding $ 75 billion in losses each year."* Also senior business executives report [Metagroup]:

- Fewer than 25% of IT projects achieved narrow, defined, project-level goals;
- Fewer than 12% fundamentally advanced the business-strategic goals of the enterprise.

A lot of IT projects fail due to a lack of understanding. The IT department accepts the requirements of the business with a requirements definition that is too simple. For more and more enterprises EA is the solution to align business with IT. EA helps to identify the people who are involved, not only to describe their requirements, but also to analyse these requirements and interpret them in the broader context of the enterprise. After that EA helps in translating the requirements into IT. EA is not only about analysis, but it is also a process between people to gain shared understanding.

Business - IT alignment is not the only reason to deploy an EA. With the help of EA processes can be rearranged in a better way or the enterprise can be governed better. In these cases EA contributes to a better understanding of the enterprise and its situation. The processes can be rearranged in such a way that they better support the business goals or the customers of the enterprise. The management can govern the enterprise, with all its processes, better because they understand the situation and the impact of their decisions. With the help of EA a clear vision can be formulated, which based on the different versions of reality that formerly resulted in a Babel-like confusion. Based on this clear vision better decisions can be made.

Better understanding is not measurable, but the following metaphor illustrates the case for architecture. If a house has to be built the principal has a design in his head. Architecture is used to communicate his idea with the contractor. With architecture the principal and the contractor can communicate, the municipality can decide if it may be build, the contractor can decide the cost price, etc. Also the contractor knows what he needs for building-materials or if the house can be build at all. This example shows that architecture is used for communication and decision-making. With the help of architecture there is a better understanding of what the principal had in mind. In the case of building a house no principal will ever let a constructor build his house without any kind of architecture. But when IT applications have to be built, it is normal to do that without any kind of architecture.

With the help of EA the different views, opinions, interests, and concerns of the stakeholders within the enterprise are taken into account. The information that has to be acquired can be seen from many perspectives. In an enterprise a lot of people are involved. They all have different views, opinions, interests, and concerns. They all look at the enterprise from a different point of view. In order to come to a better understanding of the enterprise all the different views that are relevant have to be taken into account. These relevant views on the enterprise are neither hundred percent independent nor hundred percent compatible, but together they reveal more truths about the system than either could do alone. To give a good illustration about how different views on the same object can exists, an ancient Indian tale what is called: "*The blind men and the elephant*" will be given [sheet1 Erica], [Schekkerman2003].

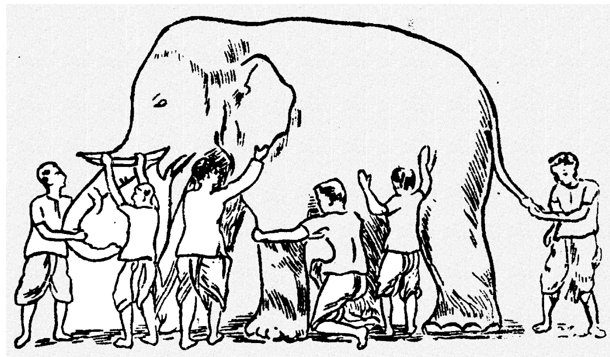


Figure 1-1: The blind man and the elephant

Six blind men lived in a village. One day an elephant came to that village, but they had no idea what an elephant was. They decided, *"Even though we cannot see it, let us go and feel it anyway."* All of them went to the beast and touched the elephant. *"Hey, the elephant is a pillar,"* said the first man who touched his leg. *"Oh, no! It is like a rope,"* said the second man who grabbed the tail. *"Oh, no! It is like a thick branch of a tree,"* said the third man who gripped the trunk. *"It is like a big hand fan,"* said the fourth man who touched the ear. *"It is like a huge wall,"* said the fifth man who pushed the belly. *"It is like a solid pipe,"* said the sixth man who embraced the tusk of the elephant. They began to argue and every one of them insisted that his image of the animal was right. A wise man came and asked them, *"What is the matter?"* They said, *"We cannot agree to what the elephant is like."* Each one of them told him their conclusion. The wise man calmly explained to them, *"All of you are right. The reason every one of you is telling it differently is because each one of you touched a different part of the animal. So, actually the elephant has all those features you felt." "Oh!"* everyone said. There was no more fight. They felt happy that they all were right.

The moral of this tale is do not assume to have the whole truth just because one part of the truth is known. By sharing the different perspectives a more complete understanding of the truth arises. Thus to understand the enterprise all the different views have to be taken into account. A different version of the same story as the blind men and the elephant is a Persian story, which is narrated by Molana Jalaledin Molavi (Rumi). It is about a group of men who discover a strange object in complete darkness. Since the storyteller is in the dark himself, he cannot provide a clue about the object. Here, all efforts to identify the object by touching its different parts prove fruitless until someone arrives with a light. The light, which in this context is a metaphor for methodology, enables them to see the whole at last. This light must be provided by a way of thinking other than analysis.

The moral of the second story is that the story becomes more complex if there is no preconceived notion of the subject. In the first story it is known that the blind men are identifying the elephant. Then it is clear what the blind men are identifying and why they compare it with a tree, a rope etc. The second story tells that to come to an understanding of something some kind of methodology is necessary.

To think about systems requires an image or concept of it: a model. With the help of models it is possible to acquire [Gharajedaghi]:

- **Information:** this is descriptive and it give answers to question that begin with words like what, which, who etc.;
- **Knowledge:** this is instructive and it is conveyed by answers to how-to questions;

- **Understanding:** this is explanatory and it is transmitted by answers to why questions.

Information, knowledge, and understanding form a hierarchy. Understanding presupposes both information and knowledge and knowledge presupposes only information. Today most enterprises focus on requiring information and knowledge, but they forget to require understanding. Understanding is to be able to explain the properties and behaviour of the system and the reason behind it. In section 3.5 the benefits of EA will be discussed and when EA acquires understanding these benefits could be accomplished. In chapter 4 the frameworks will be discussed and the what, how, and why questions are part of these frameworks. In many cases EA focuses only the information part, but that is not enough to create understanding.

This thesis is about Enterprise Architecture and especially about the selection process of an Enterprise Architecture Toolset to support understanding and governing the enterprise, which is also the title of this thesis. According to Paul Harmon enterprises often establish an EA by following seven steps. These steps are [BPT2003]:

1. **Agree on the Need:** senior management has to agree that a description of their EA is needed, before an EA can be established within the enterprise. The different approaches (from section 3.3) play an important part in the decision. The management decide the reason for creating an EA and therefore they choose an approach;
2. **Establish an Organizational Structure:** who is going to use, manage and maintain the EA? Who or which department is going to be responsible for it?
3. **Select a Framework:** a framework consists of processes and templates used for EA. A framework aligns the various levels and aspects of EA, provides a language to share, discuss and reuse and organizes cross-function traceability of arguments. It supports the establishment of an EA and they often describe specific approaches to organizing an EA [Rietveld et al. 2004 ITG 1];
4. **Select a Tool and Repository:** a tool is needed to store and relate the models and information of an EA. This has to take place in such a way that different employees can use EA within the enterprise;
5. **Organize the Existing Material:** all the existing models and information that is specified by the framework must be collected;
6. **Begin Using the Enterprise Architecture:** EA is new and managers have to learn to use it. Eventually it evolves into a key tool for the management of the enterprise. This step applies to a governance-centric approach;
7. **Extend and Maintain the Architecture:** once an EA is established, it has to be kept up-to-date. If there is a change within the enterprise, the concerned models and information in the database have to be changed. Missing models and information have to be developed.

Before the start of an EA project, the value (for the business) and the governance have to take in consideration to enlarge the chance of success. The assignment of Philips reflects only on step 4. But to select the right tool, also steps 1, 2 and 3 have to be taken into account. These steps are decisive in the process of selecting a tool. After all a tool must support the chosen approach, framework, and the people, who have to use the tool. These people have to be able to use the tool for their purposes. Philips has decided to use the

Integrated Architecture Framework (IAF) of Cap Gemini as a standard for architectural activities within Philips. This implies that the software tool should be able to support the IAF framework.

Step 1 and 2 are not clear at the moment. There is not a lot of support for the tool yet. At the moment it is not decided to acquire the tool or to use it in the future. The IAF framework is going to be used in projects if the architects consider it needed. At this stage it is not clear for everyone what the IAF framework means in practical situations and how to deal with it in a structural way.

To create a better understanding of the enterprise all the relevant views on the enterprise have to be modelled. Governance is about deciding which views are relevant and just enough views have to be considered to understand the enterprise or a part of it sufficient. These views are not independent; they are related to each other. The composition of all the different models and information about the enterprise and the relations between them is called EA. A methodology is needed to develop and manage an EA. An EA tool provides such methodology. Philips is looking for a tool that provides a formalism to develop and manage their EA in a structured way. With the tool it has to be possible to model, store, manage, and share information about EA.

This document is split up in three parts. The first part consists of chapter 2 and in this chapter some concepts and background information will be given. First in this chapter the terms system and model will be described. After that the terms enterprise and architecture will be discussed. Architecture is a well-known term in the field of construction, but in the field of enterprises it is quite new. Zachman used the term 'architecture' for the first time in a systematic way in 1987. EA is still a young profession and at this time there is limited consensus on definitions and methodology. There are a lot of similarities with architecture in construction, but there is a major difference. In construction architecture is mainly focussed on results; Enterprise Architecture it is also a process, which guides the transformation process on every level in the enterprise. In this part these terms will be described from different viewpoints, but many times the view from computer science will have the upper hand.

The second part contains three chapters that describe in general the theory of this thesis. During the theory the background information of the first part will be used. In chapter 3 the term EA will be defined. The created definition of EA will be compared with the definitions that already exist. After that, some benefits of using an EA will be described and also the scope and the different approaches of an EA will be discussed. Chapter 4 contains the theory of the EA frameworks. To create a structure and an overview in the different models and information about the enterprise an EA framework is needed. It can be used to identify the models that are missing or the pattern that has to be followed to generate the models. Also the EA framework identifies and visualizes the relations between the models and it contributes in Business - IT alignment, business redesign or a better governance. This section will describe the EA frameworks in general, but the IAF framework will be discussed more thoroughly. In chapter 5 the EA tools will be discussed. These tools are a methodological way to model, store, manage and share models and information about EA. The tools provide a formalism to create an EA in a structured way and they are the aid to generate communication within the enterprise. In this section the thoughts behind the different tools and history of them will be given.

The third part contains chapter 6 and 7 and these chapters are about the situation of Philips. Chapter 6 is an introduction to the tool selection process in which the situation of Philips will be described. It will provide a link between the theory part and the selection process of chapter 7. Chapter 7 is about the tool selection process itself. An EA tool is not a panacea that generates an EA in no time. Therefore a well-considered selection process has to take place. This tool selection process is based of theory and practical experience. The theory is from the book *Business Information Systems by Data Structures* of Jos C.J. de Heij. The practical experience is from earlier tool selection processes within Philips and a tool selection process, which was executed by Gé Schellen.

2 Concepts and background information

This chapter contains concepts and background information on which the rest of this thesis is based. In the introduction it was stated that a model is required to think about a system. In section 2.1 the terms system and model will be described. Enterprise Architecture, which is the main topic of this thesis, is a composition of the words enterprise and architecture. Therefore first of all the definition of these words will be separately given and the definition of EA will follow in section 3.1.

2.1 Systems

In the 1940s the German biologist Ludwig von Bertalanffy introduced the general system theory and Ross Ashby extended it. This theory suggests that all identifiable parts of a system are related to other parts. To understand a system it has to be considered in its totality. It is not possible to understand the whole system solely by examining its isolated parts Already in the eighteenth century the German philosopher Hegel suggested that the whole is more than the sum of its parts. In understanding the whole system, its parts cannot be considered in isolation. Thus the whole system and its mutual relationships have to be taken into consideration. The notion of the holistic view approach in understanding the relations between the parts of the whole is called in German the 'Gestalt' [McGriff].

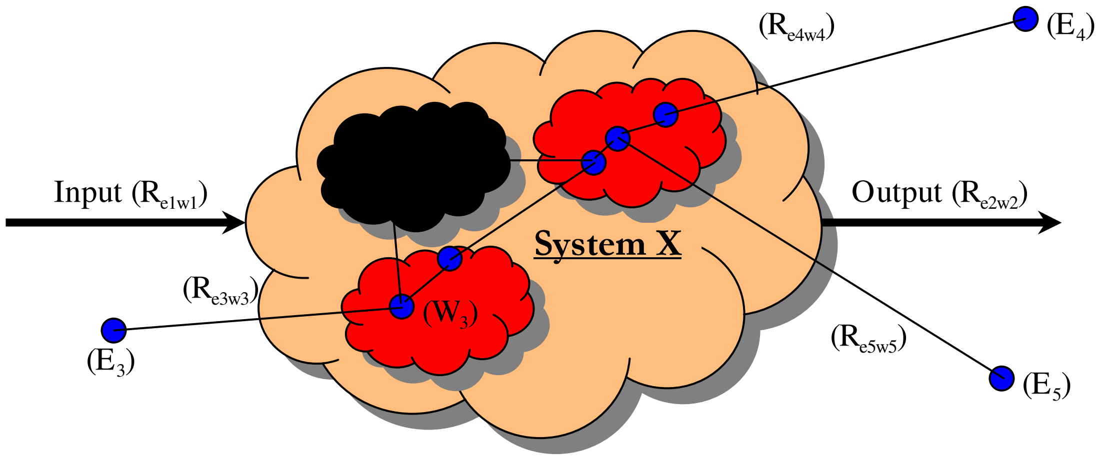


Figure 2-1: A visual representation of System X in its environment

Figure 2-1 represents a system and this system is an open system because it interacts with its environment. The internal relationships are stronger than the external relationships and that distinguishes the system from its environment. An open system, which interacts with its environment, can evolve by acquiring new properties. In many cases the environment is left out of consideration. In physics for example theories are made in ideal situations without considering the influence of the environment. But in this thesis open systems will be taken into consideration. An open system interacts with its environment and the system of Figure 2-1 interacts with its environment with an input and an output arrow. Besides these arrows the system interacts also with elements or entities in its environment. The blue circles represent those elements or entities. The relationships between them are represented by the black lines and there are some relationships with elements outside the system. The big cloud represents the whole system. The smaller clouds (the two red ones and the black one) represent the subsystems of this system. The black cloud represents a black box. It is unknown what exactly is happening in a black box, only its input and output is known. A subsystem satisfies, just like the system, the following four important characteristics of an open system [McGriff], [Rietveld et al. 2004 ITG 1], [Rogier1998]:

1. It is goal oriented (either by evolution or design);
2. It has input from its environment;
3. It has output to achieve its goal;
4. It gets feedback from the environment about the output.

These characteristics of an open systems leads to the following definition for a system, which is derived from the definitions Dictionary.com and the Interoperability Clearinghouse:

System: *"A collection of interacting, interrelated or interdependent components forming a complex whole, which is organized to accomplish a specific function or a set of functions."*

A system has to accomplish a specific function or a set of functions. That is also the first characteristic of an open system. A system is thus goal oriented what results in people that have a certain interest in the system. People who have interest in a system will be mentioned by the term System stakeholder. The definition for a System stakeholder is derived from the definition of the Northern Colleges Network Ltd.:

System stakeholder: *"A person or group within or outside the system that is involved or affected by the performance of the system."*

De Leeuw (1982) has defined a system (S) as: $S = \langle W, E, R_{ww}, R_{ew} \rangle$, in which [Rogier1998]:

- W (whole): is a collection of elements or entities that make up the system;
- E (environment): is the collection of elements or identities, which are not part of the system;
- R_{ww} is the collection of relations between the elements or identities of W;
- R_{ew} is the collection of relations between an element or identity of W and one of E.

According to De Leeuw a system can be split up in different kinds of systems and here the following two will be considered [Dimensie]:

- **Subsystem:** a subset of the collection of elements or entities that make up the system. All the relations between the elements or entities of the subset will be considered and a subsystem itself can be considered as a system. For instance if Philips is considered as a system than Philips Lighting can be seen as a subsystem of Philips.
- **Aspect system:** the whole collection of elements or entities that make up the system is taken into consideration, but only a subset of the relations is taken into account. For example the legal system of Philips is an aspect system of Philips.

A system can have different behaviours and it can be classified into three categories [Gharajedaghi]:

- **State-maintaining system:** it reacts on a determined manner to changes. The system does not learn from the past, it always reacts in the same way. Such systems have a determined behaviour and a determined goal. An example of this kind of system is a software system. With the same input it will generate the same output;

- **Goal-seeking system:** it can respond differently to one or more different events in one or more different environments and it can respond differently to a particular event in an unchanging environment until it produces a particular outcome. Such systems can choose their behaviour but they have a determined goal. An example of this system is a lower-level animal. He can try to get food in different ways in the same environment;
- **Purposeful system:** it can produce the same outcome in different ways in the same environment and it can produce different outcomes in the same and different environments. Such systems can choose their behaviour and their goal. An example of such system is an enterprise or a human being. They can choose their way to achieve their goals, but they also can change their goals.

The purposeful systems have all the capabilities of the goal-seeking systems and the state-maintaining systems. The goal-seeking systems have of course all the capabilities of the state-maintaining systems [Gharajedaghi].

2.1.1 Information Systems

In the software architecture, which will be described in section 2.3, software systems are often considered as systems that have a countable set of states. These systems, which are called transition systems, make state transitions at discrete moments. Transition systems are state-maintaining systems or goal-seeking systems. A very important type of a system is called information system. In this system the state is formed by a set of information objects. These systems are often used in computer science, industrial engineering and electrical engineering [Van Hee1994].

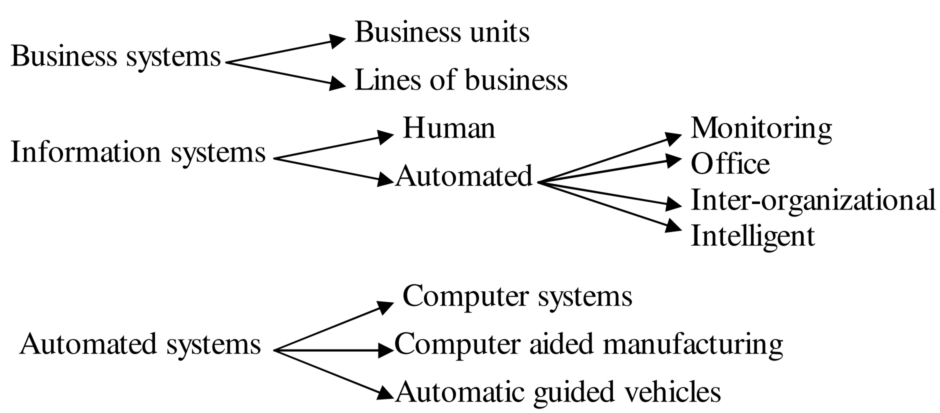


Figure 2-2: A taxonomy of discrete dynamic systems: 'human', 'automated', 'monitoring', 'office', 'inter-organizational' and 'intelligent' are all types of information system [Van Hee1994].

Discrete dynamic systems are characterized by the fact that, at each moment, they are in a state and there is a sequence of events moving the system to another state. In fact a system is always in a certain state, but an infinite number of different states can exist. Three subtypes of discrete dynamic systems are [Van Hee1994]:

- Business systems (such as a factory or restaurant);

- Information systems (whether automated or not);
- Automated systems (systems that are controlled by an automated information system).

Discrete dynamic systems are a subclass of dynamic systems, but it is a very important one. All computerized systems such as distributed databases, user interfaces, electronic mail systems and decision support systems fall within this class; so do many business systems such as factories, offices and transport enterprises. To consider these systems as discrete systems all kinds of analysis, such as simulation, can be easily executed. Discrete dynamic systems are built of two kinds of components: actors and objects. Actors are the active components in the sense that they consume and produce objects, which are the passive components [Van Hee1994].

Actors can be human beings, machines, vehicles or networks of other discrete dynamic systems. Actors perform operations on the state of the whole discrete dynamic system although they mostly consume only a subset of the tokens in a state. Objects can be material objects such as chairs and cars, or information objects like insurance policies, diagnoses, advice, designs or financial transfers [Van Hee1994].

During the analysis and design stages of the development process for discrete dynamic systems, the systems engineering is working with models of these systems. A model is in fact another system, but one that is easier to analyze or observe than the original system. But the model has so much similarity to the original system (in certain aspects at least), that conclusions drawn from the model are assumed to be valid for the original system as well [Van Hee1994].

The systems, which are discussed here, are subject to changes during their lifetime. At any moment in time the system has a certain state; at any later moment this state may have changed. The set of all states that a system may have is called its state space. A system state can be regarded as a snapshot of the system, showing all relevant details. It may often contain some details that cannot be directly observed. For example, a state may contain temporal information, such as the time at which the current state was reached or the time at which the current state will end. Another approach is to specify directly a set of allowed traces. In practice, this amounts to formulating constraints, i.e. predicates that together characterize this set. The set is prefix-closed: a prefix of an allowed trace is also allowed. One often distinguishes static and dynamic constraints; static constraints restrict the state space of a system, while dynamic constraints restrict the state transitions that can be made [Van Hee1994].

To define open systems, an open system and its environment are defined here in such a way that they form a closed system together. The combination of two open systems is called a composition. An important aspect of system composition is that the concept of communication between systems can be defined. Two systems can communicate if they are components of a compound system. When describing a complex system, it is often decomposed into less complex systems called components. The original system is then a compound system, the composition of the components [Van Hee1994].

In this thesis two different approaches were presented to describe systems. The first approach is from the general system theory and this approach will mostly used to describe complex systems like enterprises from a business point of view. In the second approach a system is considered as a discrete system and a discrete system is always in a certain state. The number of possible states marks its state space. These kinds of systems are mostly be used by the IT department for information systems. EA has to deal with all kinds of systems and therefore the definition that is introduced in this chapter is a general definition that is valid for both kinds of systems. To gain shared understanding between people systems have to be considered from different perspectives. In this chapter two possible perspectives were given to illustrate different viewpoints considering systems.

To understand a system there are three things that have to be examined [Gharajedaghi]:

- **Structure:** the way in which the work is divided among the parts of the system and how the parts influence each other (relationships). In order to understand the system its structure can be understood only if observed in the functioning of the system. To do this analysis is needed to understand the individual parts, and synthetic thinking is needed to understand the parts in the whole system;
- **Processes:** this is about understanding which parts of the system respond to their environment. Processes are responsible for future states of the system and which outcomes the system has;
- **Functions:** to understand the functioning of a social system the cause effect relationship is inadequate and a producer-product relationship is required [Singer]. A producer is only necessary, not sufficient, for its product. Therefore he does not explain its product totally. This makes it possible to treat choice, hence purpose, as an objectively observable property of system behaviour. Moreover, since a producer is not sufficient for its product, reference to its environment is required to explain its product. Therefore explanation of a social system is environment-full rather than environment-free.

Without understanding its structure, processes and functions, it is not possible to understand the enterprise. For example, a lack of understanding may result in the misalignment between the business and IT. By seeing the enterprise from different viewpoints, its structure, processes, and functions can be better understood. This results in IT projects that are better adjusted to the situation of the enterprise [Schekkerman2003].

To study the behaviour of a system and its effect on its environment, by experimenting in reality, is often not desirable. That is because of the time it will take, the costs involved, or the changes it might cause to the system and its environment. Another approach is to examine the structure, processes and functions of a system by simplifying the description of reality by leaving out irrelevant aspects. That can be done with the help of a physical or mental construction. These simplified constructions are also known as models. By examining a model from different perspectives, using it in discussions and simulate possible events, a better understanding and control of the complex reality is achieved. Models are used for various reasons [Van Hee1994]:

- **Design:** used as a blueprint of a system that has to be constructed. In this case it should be easy to map the building blocks of the model onto existing or realizable components;
- **Analysis:** used to analyze a system, for instance performance or reliability. If system exists one could observe behaviour;

- **Specification / documentation:** used to document the functionality of the system, i.e. the model describes the behaviour of a system in an abstract way. Model can be used as a specification of the system.

2.2 Enterprise

An enterprise is a system and in the previous section it was stated that an enterprise is an example of a purposeful system. In this section a definition of an enterprise will be given. This definition is based on the definition for a system.

Enterprise: *"A whole of organized people and means intended to create, with the help of processes, products and services where its stakeholders are willing to pay for."*

The customers of the enterprise are also stakeholders of the enterprise and they can directly or indirectly pay for the products and service created by the enterprise. But not always the customer are willing to pay. Also other stakeholders, like shareholders can pay for products or services. For example by investing in a new enterprise which produces goods that are not sold yet. The definition for an enterprise is very general and it has therefore a wide interpretation:

- An enterprise does not need to have a profit motive;

The government is an enterprise too and its customers are paying for it by taxes;

- A sports club is also an enterprise;
- A division of an enterprise can also be considered as an enterprise.

Philips is of course an enterprise. But also its Production Divisions (PD's) are enterprises of their own. They all produce products where its customers are willing to pay for. The research division of Philips is an enterprise where its customers are indirectly paying for. Customers are paying for end products, but the price includes a compensation for the research and development of the product.

2.3 Architecture

The concept of architecture is becoming more and more important in enterprises. Some important reasons are:

- The whole society, including the enterprises, is becoming rapidly automated. In all kinds of areas human work is supported or even replaced by information systems;
- The growth in information systems results in a world that is more complex so it becomes more difficult to create security adequately;
- The customers are becoming more demanding. To deal with these problems a structural approach is required;
- The design and realization of changes in an already complex whole. Enterprises are already complex systems and it is very difficult to make changes in an enterprise in such a way that the enterprise can function in the right way in the future.

This structural approach has been part of the world of construction for centuries. There it is known as 'architecture'.

In this section the term architecture will be defined. There are many definitions for architecture that play a role in enterprises like Business Architecture, Process Architecture, Information Architecture, Application Architecture, Software Architecture, Technical Architecture, and Infrastructure Architecture [Rijsenbrij et al.].

The most important development of the past few years, is the standard of the Institute of Electrical and Electronics Engineers, Inc. (IEEE). They have proposed a definition for architecture, which is broadly accepted. Through its members, the IEEE is a leading authority in technical areas ranging from computer engineering, biomedical technology and telecommunications, to electric power, aerospace, and consumer electronics, among others. The IEEE Standard 1471-2000: IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, which is called IEEE 1471 from now on in this thesis, is [IEEE]:

Architecture: *"the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution."*

Now this definition will be explained by defining and explaining all the elements that play a role. At the end some general remarks on IEEE 1471 will be given and a model will illustrate the definition. In section 2.1 a definition of a system and a system stakeholder were given. The definition of IEEE starts with: *"the fundamental organization"* and contains *"environment"*. The fundamental organization means essential, unifying concepts and principles. This 'internal structure' of the enterprise cannot always be clearly seen from the outside and especially in the case of complex organisations it cannot be described completely. Therefore the definition contains the word *"embodied"*. The environment is the developmental, operational, programmatic, … context of the system. A system inhabits an environment and the environment has influence on the system. The interaction with the environment is very important for the open systems, which is described by characteristics two, three, and four in section 2.1. These characteristics are all about interaction. The environment can include other systems that interact with the system of interest. To understand a system's most fundamental characteristics, it is crucial to know how a system relates to and is embedded in its environment [AWG1998], [FAQ], [Hilliard].

IEEE 1471 ends with *"principles guiding its design and evolution"*. The definition of a principle that will be used in this thesis is [Cap]:

Principle: *"A expression of an idea, a message (culture) or value that comes from corporate vision, strategies, and business drivers, experience or from knowledge of a subject."*

So the principles can have influence on the system from the moment of its design and in every phase after that. A principle will [Cap]:

- Underpin the investigations made when looked at architectural options;
- Be used to justify the decisions that were made about the components in the architecture;
- Ensure that the architecture that is defined, is consistent and that there is no wander from path to the goal.

Here it is stated that principles can have influence on a system because it is also possible that organisations develop without explicit principles. Examples of principles can be found in section 4.4.3.

In the definition of the term 'system' (given is section 2.1), a system has to accomplish a specific function or a set of functions. In the terminology of IEEE the goal of a system is called its mission. The definition of a mission, which will be used in this document, is [AWG1998]:

Mission: *"a use or operation for which a system is intended by one or more stakeholders to meet some set of objectives."*

IEEE 1471 standardizes conventions on AD's and an AD is a document or some other artefact that defines an architecture. An architecture is a conceptual thing whereas an AD is a concrete artefact. This does not mean that every architecture has an AD because an architecture can exist without any representation of it [FAQ]. The definition of an AD, which will be used in this thesis, is [Land]:

Architectural Description: *"A collection of products to document an architecture."*

IEEE 1471 is an organizing framework intended to supplement frameworks like Zachman or IAF by providing specific content requirements on AD's. IEEE 1471 does not define a minimal required content of an AD. It does not specify the format or media for an AD, so an AD is notation-independent. An AD does require that the system's stakeholders and their concerns will be identified. It identifies them and it addresses all relevant (or considered relevant by the person who is responsible for the EA) stakeholders' concerns [Hilliard], FAQ]. The definition for a concern is derived from the definition of Wordnet:

Concern: *"Something that interests the system stakeholder because it is important to him or it affects him."*

All the people, who are in any kind involved with the system, are system stakeholders and they all have their own particular concerns. Typically concerns include: security, integrity, functionality, etc. An AD consists of one or more views and it documents any known inconsistencies among the views it contains [Hilliard]. The definition of view that will be used in this thesis is [Land]:

View: *"A representation of a whole system from the perspective of a related set of concerns"*

A view may contain one or more architectural models and therefore it is possible that a view utilizes multiple notations. Each concern is addressed by an architectural view. Also every view corresponds to exactly one viewpoint. The definition of a viewpoint that will be used in this thesis is [AWG1998]:

Viewpoint: *"A pattern or template from which to construct individual views. A viewpoint establishes the purposes and audience for a view and the techniques or methods employed in constructing a view."*

A viewpoint is a way of looking at a system and a view is what is seen looking from the chosen viewpoint. A view is a collection of models, which together represents the system with respect to a set of related concerns. A viewpoint captures the rules for analyzing and constructing a particular kind of view. A viewpoint establishes the conventions by which a view is depicted and the architectural techniques or methods to create and document that view. It determines the languages to describe the architecture and any analysis technique that can be applied to these descriptions. The languages and techniques are used to get results that are relevant to the concerns addressed by the viewpoint. An AD selects one or more viewpoints to express the views it contains. The selection of the viewpoints is based on the consideration of the system's stakeholders (to whom the AD is addressed) and their concerns. A viewpoint can be reused and a viewpoint, which is already defined, is referred to as a library viewpoint [AWG1998], [FAQ].

A view consists of one or more models in the definition of IEEE 1471. In their opinion a viewpoint has only one view and that view consists of a number of models. Every model is developed by using the methods that are established by its associated viewpoint. A model can participate in one or more views. The definition of a model that will be used in this thesis is [AWG1998]:

Model: *"An approximation, representation or idealization of selected aspects of the structure, behaviour, operation or other characteristics of a real-world process, concept or system."* (IEEE Std 610.12-1990)

Models are very important in creating an AD (see also section 2.1). With the models it is possible to create an AD and thus to describe an architecture. An architect has to select a modelling language to be able to create and share the models. The relations between components in the enterprise are represented by the relationships between the different models, so the models are not independent of each other [Delen et al.]. There are two types of relationships between the components in one view. These are [Van Hee]:

- **Interaction relations:** relations like 'uses', control flow and dataflow;
- **Structural relations:** relations like is-part-of and is-specialization-of.

There are many types of relations between the different views and between components of different views. Examples of these relations are: is-realized-by, is-implemented-on [Van Hee].

An AD includes also a rationale for the selection of the architecture. The AD must contain, on every architectural level, a rationale for the architectural decisions and choices made by the architect [AWG1998]. The definition of a rationale is [Bruegge & Dutoit, 2000]:

Rationale: *"the justification behind decisions. It includes the issues that are being resolved, the alternatives being considered and discarded, the criteria used to evaluate alternatives, the assessments of alternatives against criteria, the argumentation, and the decision."*

The selection of the system's stakeholders, their concerns and viewpoints is the responsibility of the architect or the owner of the EA. An architect is a role of a person, team or organization that is responsible for the activities of defining, maintaining, improving and certifying proper implementation of an architecture. The architect has to construct the views and he has to execute the creation and maintenance of the architecture. He has to choose a suitable viewpoint after identifying a system stakeholder and his concerns. Unlike the system's stakeholders and views, the viewpoints are not system specific. This makes it possible for an architect to reuse viewpoint descriptions [AWG1998].

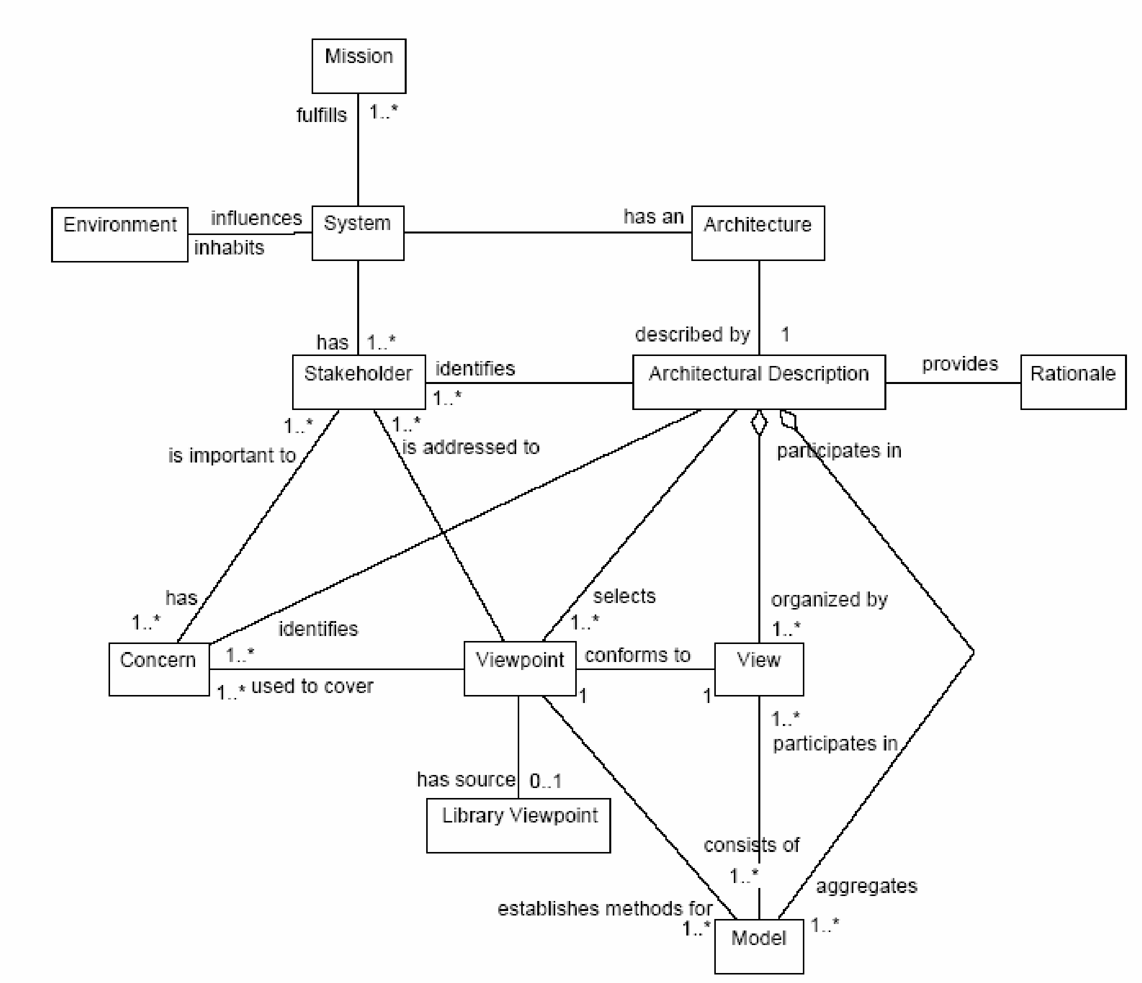


Figure 2-3: IEEE 1471 Conceptual Framework [AWG1998]

Figure 2-3 is the conceptual framework of definition IEEE 1471. This figure applies to all the different systems and architectures. To give a better understanding of how those concepts, which are discussed before, relate all these concepts will be drawn in Figure 2-3. A system has always one architecture and this architecture can be described by one AD. An AD can consist of many views, models, and viewpoints. The rectangles represent the classes of objects. The lines connecting two objects represent the relationship between these objects. The relationship has two roles (one in each direction) and a role can be named. In Figure 2-3 a role is a one-to-one role unless otherwise noted. For example: a stakeholder has one or more concerns and a concern is important to one or more stakeholders. The notation in Figure 2-3 is from the Unified Modeling Language (UML) [AWG1998], [UML].

According to the definition IEEE 1471, an AD must include the following elements [AWG 1998]:

- Identification of the system stakeholders and their concerns judged to be relevant to the architecture. The decisions are made by the 'owner' of the architecture;

- Definitions of each viewpoint which has been selected to organize the representation of the architecture;
- Representation of the architecture, organized into views;
- A record of all known inconsistencies among the AD's required constituents;
- Rationale for selection of the architecture.

2.3.1 Evaluation of IEEE 1471

Now a short conclusion on IEEE 1471 will be described. The formulation of IEEE 1471 is very subtle, that is because the formulation is an abstraction from the examination of a wide range of best practices in these methods. IEEE 1471 does not only refer to the overall structure of physical components that make up a system, but it also identifies the process aspects of an architecture. While physical structure is often a fundamental aspect of a system, it is not necessary. Also it does not require an AD to cover the system's boundary. The main advantage of this definition is that it is clear and concise [FAQ], [Greefhorst], [Rietveld et al.]. Besides that IEEE 1471 has the following advantages:

- It defines that every system has an architecture, even if the system does not have an AD or if the system was not designed but grew or emerged. Therefore an AD can be created for every existing system;
- It links directly to the systems theory, so concepts and definitions can be reused;
- It defines that architecture takes the context of the system of interest into consideration;
- It defines that AD's are structured to meet the needs of the multiple stakeholders of the system, including, but certainly not limited to, the builders of the system;
- The concerns of multiple stakeholders are most readily addressed by AD's constructed with multiple views of the system, with each view covering an identified set of system concerns;
- The conditions on the completeness, well-formedness, and analyzability of views are made explicit to readers of an AD in an architectural viewpoint (just as maps have legends next to them).

IEEE 1471 does not require [Helliard]:

- A particular architecture description language;
- A required view or model;
- A required formal consistency or completeness criteria.

2.3.2 Software Architecture

Now a specific architecture will be considered and in this thesis Software Architecture is selected as illustration. In this field, which is a relatively new discipline, there is still not a uniform standard. Software Architecture is a high level design of a system and it describes the most important characteristics of a system [Greefhorst]. An important role of an architecture is that it can be used to verify and validate properties of the system. Kazman defines software architecture as [Kazman et al.]:

"The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them."

This defines software architecture as a set of entities with relationships between them. It leads to the following implications [Kazman et al.]:

- **Architecture defines elements:** architecture embodies information about how the elements relate to each other. An architecture is foremost an abstraction of a system that suppresses details of elements that do not affect how they use, are used by, relate to or interact with other elements;
- **Systems can and do comprise more than one structure:** also no one structure holds the irrefutable claim to being 'the architecture';
- **Every software system has an architecture:** because every system can be shown to be composed of elements and relations among them, every system has an architecture. Even though every system has an architecture, it does not necessarily follow that the architecture is explicitly described. Unfortunately, an architecture can exist independently of its description or specification, which raises the importance of architecture documentation and architecture reconstruction;
- **The behaviour of each element is part of the architecture:** insofar as that behaviour can be observed or discerned from the point of view of another element. This behaviour is what allows elements to interact with each other, which is clearly part of the architecture. This does not mean that the exact behaviour and performance of every element must be documented in all circumstances; but to the extent that an element's behaviour influences how another element must be written to interact with it or influences the acceptability of the system as a whole, this behaviour is part of the software architecture.

The implications are the same as those from the IEEE 1471 definition. But in this definition Software Architecture is only a description of a system's structure. But an architecture has also to identify the process aspects of it. This definition is partly suitable for architecture, because it only reflects the description of a system's structure.

Now the most relevant developments in Software Architecture will be discussed. These will be related to IEEE 1471. In Figure 2-4 it is indicated with the help of numbers where certain developments can be located. The developments are [Greefhorst]:

1. **Architectural frameworks:** frameworks offer standard viewpoints (the library viewpoints), which are ordered in one or more dimensions. Examples of frameworks are Zachman and IAF and these frameworks will be discussed in chapter 4;
2. **Architectural patterns:** these are standard solutions for problems that often occur. These patterns can be seen as standard models, or part of models. Architectural patterns describe the collaboration of components at a macro level. In this case the components are seen as black boxes for which only their input and output is known;
3. **Architectural descriptions:** there has been a lot of research to Architectural Description Languages (ADL's) that can describe the architecture. These languages offer a formal architectural description that can be expressed in interfaces, relations, and functionalities of components;
4. **The evaluation of the architecture:** is about the quality of the architecture. Quality aspects can be seen as specific concerns and the evaluation of architecture can be seen as an instrument to measure to which extent the architectural descriptions satisfy the concerns of the different stakeholders. By evaluating the architecture in an early stage, the project can be adjusted and other solutions can be examined;

5. **Product-line development:** is aimed to develop the basics of similar systems once. The development processes are being separated in domain engineering processes, and application engineering processes, in which the collaboration development happens in the first group of processes and the system specific development in the second;
6. **Architectural construction:** extracts the architectural information from existing systems. In terms of IEEE 1471 it is extracting the architecture from the system and translating it in an architectural description. The internal processes, input, and output of the system are used to describe that system.

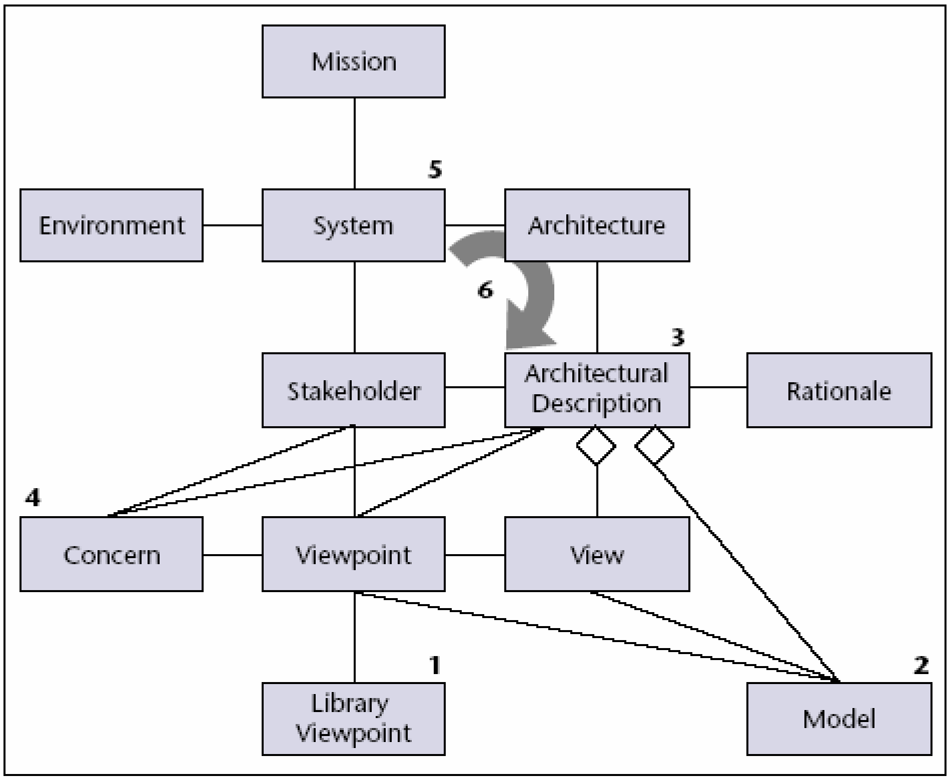


Figure 2-4: Developments in Software Architecture projected in the IEEE 1471 Conceptual Framework [Greefhorst]

3 Enterprise Architecture

Enterprise Architecture (EA) exists since the early eighties of the last century. For more and more enterprises EA is the solution to align business with IT. But the meaning of EA is unclear for many enterprises. According to Gary A. Bolles Enterprise architecture is the most-used and least-understood concept in IT. There are many papers written about EA in the world today, but it seems that every time a new paper is published, there also appears a new definition of EA. So what is EA exactly? To answer this question and to get more feeling for the concept of EA, this chapter contains five different parts. In section 3.1 the definition of EA will be introduced and explained. This definition will be used through this whole thesis. Section 3.2 will give an overview of the As-is architecture, the To-be architecture and the transformation process. Section 3.3 describes the different approaches of EA. The approaches that will be discussed are the IT-centric, business process-centric and governance-centric approach of EA. Section 3.4 is about governance because EA has a decision-making aspect in it. Section 3.5 sums up and describes the benefits of using an EA. Section 3.6 is about Enterprise Information Systems. It seems that these systems like ERP systems have found a way to align business processes with IT systems. This alignment will be discussed together with how EA should be used in combination with such systems. The last section of this chapter is about a vision for the future, what is needed to achieve that and how EA can contribute towards that end.

3.1 Definitions of Enterprise Architecture

There is an increasing belief that EA is very important, but there is still not a general consensus about the definition of EA. In section 3.1.1 a new definition will be introduced and the intention is to introduce a general definition. The definition of Enterprise Architecture will be based on the theory from chapter 2. In section 3.1.2 examples of different architectures will be given and how the independent architectures adapt into an EA. Section 3.1.3 compares the new definition with existing ones.

3.1.1 The new definition of Enterprise Architecture

An enterprise is a system that is under constant pressure to optimize its performance and to adjust to a changing environment. The development of technology enabling a global economy has resulted in larger and more complex enterprises and yet they have to be more agile than ever in order to deal with the increased speed in which changes in the environment occur. To optimize its performance, the enterprise has to be understood in the way it functions and behaves in relation to outside influences. Understanding the enterprise is the reason to create an EA. Therefore its definition should be formulated in such a way that EA contributes in understanding the enterprise. So the definition has to deal with the three conditions (Structure, Processes and Functions, which were mentioned in section 2.1) that are needed to understand a system. Therefore the definition of an EA, which will be used in this thesis, is:

Enterprise Architecture: *"the fundamental organization of an enterprise embodied in its different architectural descriptions, their relationships to each other and to the environment and the principles guiding its design and evolution."*

Although the name gives another assumption, EA itself is not an architecture. EA is a meta-architecture that consists of a set of AD's from all the architectures in the enterprise. Together they help to understand the enterprise just like an architecture helps to understand a system. Besides a set of architectures EA contains all kinds of rules on the architectures (e.g. who its owner is) and these rules are taken down in the definition as principles. The formulation of this definition is also meant to be very subtle. This definition should be suitable for all kinds of Enterprise Architectures and it is derived from the IEEE 1471 definition for systems. Another purpose of the definition is that it is not limited by its vocabulary or a specific viewpoint. Therefore this definition is neutral with respect to specific viewpoints. In the rest of this section this definition will be explained [AWG1998].

The fundamental organization from the definition means essential, unifying concepts and principles of the enterprise. This is about the internal structure of the enterprise and that is one of the three things that are needed to understand an enterprise. The criterion for something being 'fundamental' depends on what needs to be achieved. Because an enterprise is a very complex system, a complete description cannot be created [Rietveld et al. 2004 ITG 1].

The definition of an enterprise from section 2.2, makes it possible that an enterprise is a division of an enterprise. Therefore a division of an enterprise can also have its own EA. An EA is embodied in the different architectures of the enterprise. The EA of Philips can consist of all the EA's of the different PD's together with architectures on corporate level. Because all the architectures together describe the complete enterprise, a better understanding of the enterprise is possible. In the definition *"different architectural descriptions"* is used instead of:

- *"Architectures"* because every system has an architecture and an EA describes all the systems of the enterprise, so an EA consists of architectural descriptions;
- *"Architectural descriptions"* to emphasize that an EA consists of several different types of architectural descriptions (two or more) and not only a set of Business Architectural descriptions or a set of System Architectural descriptions;
- *"Elements"* or *"components"* to define that the elements of the meta-architecture can only be architectural descriptions besides relationships and principles.

The relationships in this definition are the relationships between the AD's. These relationships are the primary added value of an EA. An architecture is the highest-level conception of a system in its environment. In fact these links exist between the components of the different AD's. These links are very complex and comprehensive and a link is a direct joining of two components. Because EA is of a higher abstraction level the links are between the AD's. The relationships between two components in the enterprise are about the processes and functions of the enterprise. There are also relationships between the principles of the different architectures. The principles are the key in finding relationships and they can be transformed and translated into other principles. So a business principle can lead to an IT principle (and the definition of EA shows that there are relationships between principles). An example of a business principle is: *"All employees have to access data, without special software, all over the world."* The matching IT principle is: *"The tool that supplies the data must be web-based and can be accessed with a browser."*

An enterprise interacts with its environment and that relates to the characteristics two, three and four of an open system in section 2.1. It inhabits its environment and its environment has influence on the enterprise. Because an enterprise is an open system, the relationship between the enterprise and its environment has to be taken into account. These relationships are in fact the interactions between the elements from the enterprise with the elements in its environment. These relationships are described by the different architectures of the enterprise.

The last part of the definition is: *"the principles guiding its design and evolution"*. These principles are the general principles of the enterprise. They apply to all the architectures in the enterprise. For example: *"Reuse before buy before make"*. Besides these general principles there are also principles that are specific for an architecture; these principles only apply to that architecture. In section 2.3 the definition of a principle is given and principles are expressed in rules and guidelines for the enterprise. They also describe the things that need to be achieved; the requirements are the way in which they can be achieved. These requirements are the missions (see section 2.3 for the definition of a mission) of the different systems, which are described by their architectures. In section 2.3 every system has a mission, which is the goal of the system. Every system has to fulfil one or more missions. In section 4.4 it will be discussed that there are relationship between principles and requirements. The example of a business principle, which was already used, is: *"All employees have to access data, without special software, all over the world."* Two examples of requirements that can be derived from this requirement are:

- An employee can only access data wherefore he is authorized;
- An employee has to have access to the history of the client.

The previous chapter contains four characteristics of an open system. The first characteristic is that every system has a goal. The enterprise itself is an open system, so an enterprise has a goal. The general goal of a social system is development. Development is the process in which individuals increase their abilities and desires to satisfy their own needs and legitimate desires and those of others. An enterprise has to develop and adapt to its changing environment in order to survive. The way an enterprise wants to develop is taken down in its general principles. Thus an enterprise is a purposeful system, not a goal-seeking system [Gharajedaghi].

In Figure 3-1 there are four kinds of systems, which are already described in chapter 2. These systems are the enterprise systems, information systems, subsystems, and aspect systems. The arrow between these systems and the entity system is an ISA relationship. It means that it means that in the context of this a system is an enterprise, an information system, a subsystem or an aspect system. A system can consist of a number of systems. An enterprise can consist of a number of enterprises. Philips consists of a number of enterprises such as the product divisions Semiconductors or Lighting. Because an enterprise consists of a number of systems, the EA is the aggregation of all the AD's of those systems.

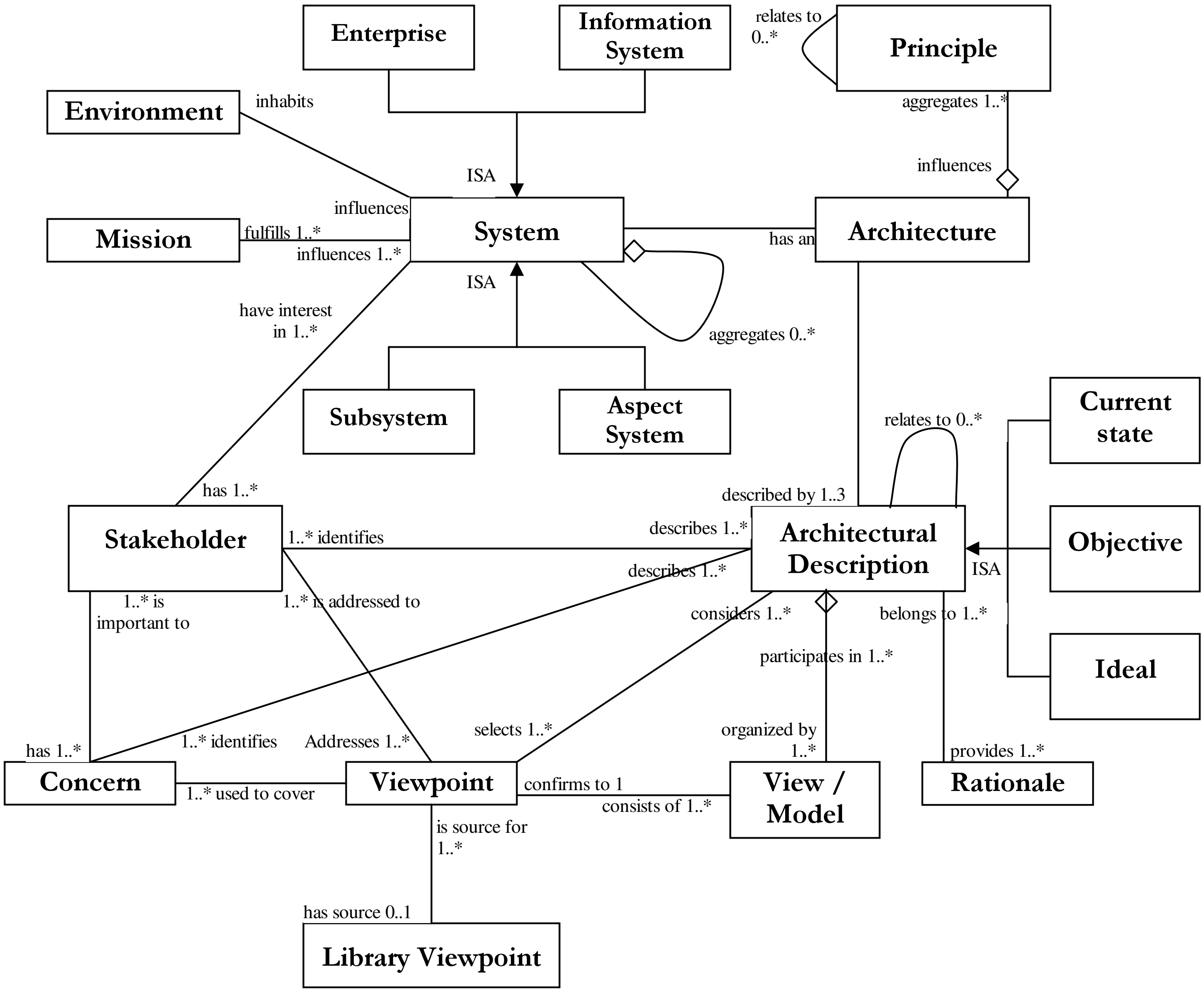


Figure 3-1: The Enterprise Architecture

Related to the time dimension there are three kinds of AD's, which are the current state, the target state, and the ideal state. The AD of the current state is a blueprint of the enterprise at the moment. The AD of the ideal situation is a blueprint of the perfect enterprise. This ideal situation would be realized if there were no technical, organization, and financial or other constraints. For the longer term this is the situation that the enterprise should aspire for. An AD of the target state is the blueprint of the enterprise that can be realized within a specific time horizon. This target should be feasible and create enough added value to compensate the costs of the change. The IAF framework, which will be discussed in section 4.4, also contains these three kinds of AD's. Figure 3-1 is an extended version of Figure 2-3. Compared with the IEEE 1471 description the objects 'Model' and 'View' are merged into one object here. Because a view has exactly one viewpoint, every model can be considered as a view that can be described by one model from that viewpoint.

3.1.2 Examples of different architecture descriptions

There are a lot of architectural descriptions that can appear in an EA. Examples are: Business Architecture, Organization Architecture, System Architecture, Application Architecture, Software Architecture, Information Architecture, Governance Architecture, Technology Infrastructure Architecture, etc. All these architectures could be defined in the same way as the definition of IEEE 1471. For example Business Architecture can be defined based on the definition of IEEE 1471 as [Rietveld et al.]:

Business architecture: *"the fundamental organization of the enterprise embodied in its business domains, their relationships to each other, and to the environment, and the principles guiding its design and evolution."*

Every definition of an architecture can be derived from definition IEEE 1471 by filling in the three blanks in the following definition:

… Architecture: *"the fundamental organization of … embodied in its …, their relationships to each other and to the environment and the principles guiding its design and evolution."*

In section 4.4.1 the six architectures of the IAF framework will be defined in this way. Every architecture has principles, mutual relationships, and relationships with its environment. Section 2.3 described that an Architectural Description consists of one or more models and examples of these models are: organization charts, diagrams, tree-models, etc. According to the definition of a model in section 2.3, plain text is also a model because it is also a representation of selected aspects of the structure, behaviour, operation or other characteristics of a real-world process, concept or system.

3.1.3 Other definitions of Enterprise Architecture

To get a better understanding of EA a definition of EA was given in this thesis. To formulate that definition existing definitions of EA were reviewed. In this section four of these definitions are evaluated in comparison with the new definition. During the comparison the strengths of the new definition will be discussed.

The first definition is from the Institute for Enterprise Architecture Developments:

"Enterprise Architecture is about understanding all of the different elements that go to make up the enterprise and how those elements interrelate."

"A good definition of ***'enterprise'*** *in this context is any collection of organizations that has a common set of goals/principles and/or single bottom line."*

"A good definition of ***'elements'*** *in this context is all the elements that enclose the areas of People, Processes, Business and Technology."*

(Institute for Enterprise Architecture Developments [IFEAD])

This definition has the same goal as the new definition: to understand the enterprise. But this definition does not take the environment of the enterprise into account. The definition of elements is not specific enough and there is no specification on how to understand the elements. Also there is not a reference to the architecture.

The next definition is of the Railroad Retirement Board:

"Enterprise Architecture (EA) is a 'blueprint' of an organization's business processes and the information systems and technology needed to perform those processes efficiently."
(Railroad Retirement Board, Bureau of Information Services [RRB])

This definition takes three architectures into account: the Process Architecture, the Information Systems Architecture and the Technology Architecture. But it ignores the other architectures and it does not take the environment and the relationships into account.

The next definition is of Maarten Leurs:

"An enterprise architecture consists of:

- *A set of equally important models of the fundamental organization of the enterprise and the interaction with its environment. The models are expressed in a modelling language (often partly a diagram technique) that, often support hierarchical decomposition to deal with components;*
- *Relationships between the models, satisfying rules that guarantee consistency and completeness;*
- *Principles governing its design and evolution.*

The models cover at least the following aspects of an enterprise:

- *Goals and strategy;*
- *Business processes;*
- *Business objects;*
- *Organization structure;*
- *Information systems (hardware and software)."*

This definition is very similar to the new one. The second part is an addition to the first part (the actual definition) and that does not belong to the definition for an EA. Every enterprise can decide which models are definitely needed to develop their EA. But this definition does not consider the relationships between the principles. There are only relationships between the models and there are no relationships between the principles or between the principles and their requirements (the mission of a system). Also in the definition of EA, which was introduced in section 3.1.1, the models do not have to be equally important. Only if you want to achieve a complete description of an EA you need all the models and from that perspective they are equally important. But it is impossible to generate a complete description of an enterprise because there are so many views and they are continuously changing. EA is used to reach a certain goal and to reach that goal not all the models are needed, nor are they equally important. For example, to rearrange the technical infrastructure an IT-architecture is needed, but to change business culture an IT-architecture is hardly relevant. Thus to reach one goal a certain model may be essential and to reach another goal the same model may be irrelevant.

The last definition is of the Global Enterprise Architecture Organisation:

"Enterprise Architecture provides, on various architecture abstraction levels, a coherent set of models, principles, guidelines, and policies, used for the translation, alignment, and evolution of the systems that exist within the scope and context of an Enterprise."

(Global Enterprise Architecture Organisation [GEAO])

This definition consists also of the models, principles (in this thesis it is considered that guidelines and policies are part of the principles) and their environment, but it does not contain the relationship between them. In fact, as mentioned before, the relationships are the added value of an EA.

3.2 The process aspect of an Enterprise Architecture

Besides giving a description of the enterprise, EA has also a process aspect in it. EA may support the transition from the current situation (the As-is architecture) up to a more efficient and effective situation (the To-be architecture). In section 3.1.1 it was explained that a To-be architecture could be split up into an ideal situation and a realistic situation. The movement from an As-is architecture towards a To-be architecture will take place in three steps (see also the illustration in Figure 3-2). The first step is to describe the As-is situation and the second step is to define the desired (To-be) situation. The To-be situation can be split up in an target state and an ideal state. The third step is the transformation process from the current to the target situation, which is captured in a transition plan [Bruijne et al.2004], [DIS].

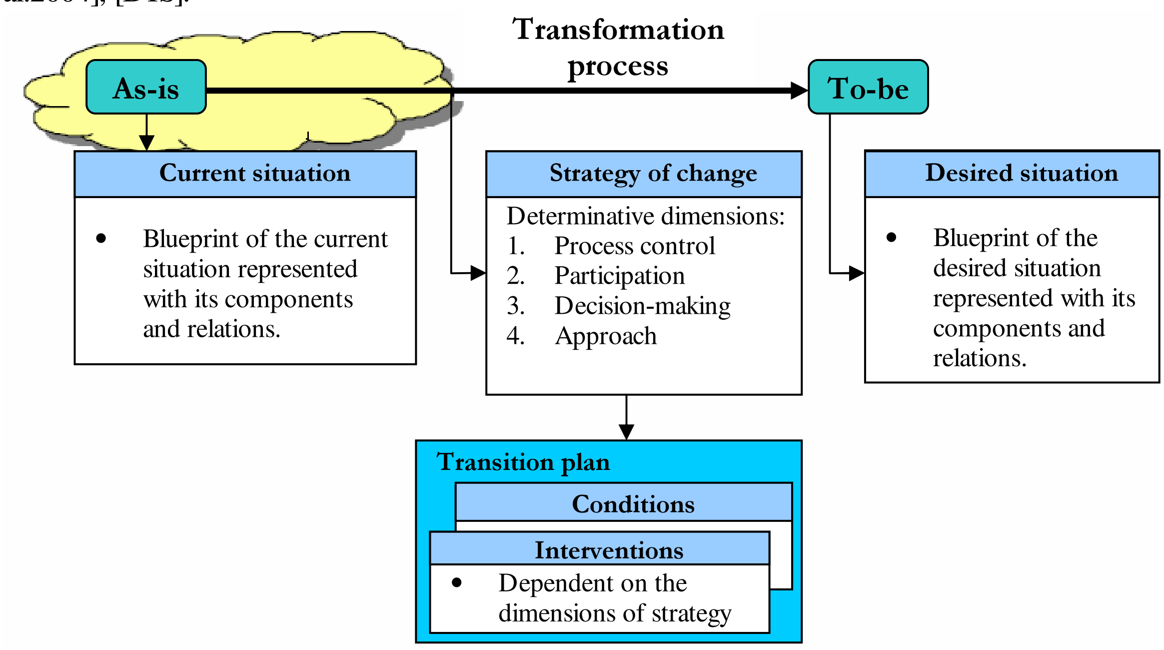


Figure 3-2: The three steps of Enterprise Architecture [Bruijne et al.2004]

The As-is architecture

The As-is architecture represents the current state of an enterprise. The As-is architecture is a blueprint of the current situation in which the components and the relations between them appear. The As-is architecture has to be made first to know how the current situation is. This architecture may reveal inadequacies like inconsistencies or inefficiencies.

The To-be architecture
The To-be architecture represents the ideal and target state of the enterprise. It is a blueprint of the ideal and target situation in which the components and the relations between them appear. This architecture represents these situations for the enterprise. The ideal situation is the situation that should be achieved. But it is not possible to achieve this situation on a short notice. Therefore the target situation is introduced and this is the situation that can be achieved in the near future. A blueprint of the ideal situation should be made just to make sure that the target situation will not interfere with the achievement of the ideal situation somewhere in the future.

The transformation process
The transformation process contains three parts: strategy of change, intervention and the transition plan. The transition plan contains all the thematic aspects of the transformation process and is a tool to evaluate the interventions on a set of conditions and a timetable. The transformation itself takes place by interventions and the strategy of change is the guideline to select the most optimum interventions. The strategy of change is based on the current situation, desired situation and the four determinative dimensions. These four dimensions are [Bruijne et al.2004]:

- **Process control:** the transformation process can be top-down or bottom-up;
- **Participation:** the transformation process can be directed to support the architects by creating an EA or it can be directed in making the architectures;
- **Decision-making:** the decision-making can be based on consensus or the decisions can be made on the top;
- **Approach:** the EA can be created and used based on different approaches, see also section 3.3.

These four dimensions are used to give a better understanding about what happens and which questions are playing a role in the transformation process.

3.3 Three different Enterprise Architecture approaches

There are different approaches of EA and they represent the different views, goals and users of EA. The following description of EA takes that into consideration [BPT2004]:

> *"… there is no correct definition of the term 'enterprise architecture.' We will continue to use it in BPTrends in both its common senses—as a description of how all the resources of an organization can be structured by the organization's business processes, and as a description of how all of the organization's computer resources are organized."*

The quotation shows that there are two descriptions of EA: one with a business point of view and one with an IT point of view. These two views are the most common ones. In this chapter, the business process-centric and the IT-centric EA (including data, information and information-systems) will be described. Also a third approach will be given and this is the governance-centric EA. There is not a best approach, but an enterprise has to select an approach that suits the enterprise to achieve its obtained goals. The approaches are different and have different objectives because managers in the enterprises may have different goals and different views. The different approaches are based on different views, frameworks, and users.

	IT-centric	Business process-centric	Governance-centric
Objective	Business - IT alignment	To manage and improve business processes	To understand the business better, to be able to govern it better.
Provide	An overview, from an IT perspective, of all the different IT and business models and resources and how they relate	An overview of all the different business processes and the relations between processes and the environment	An overview of all the domains within the enterprise described from different viewpoints
Focus is on	IT systems and their architecture	Business processes	Communication between Business and IT; collaboration between all the divisions and managers
Initiators	IT managers	Business managers	All the managers of the enterprise
Users	IT department	Business departments	All departments

Table 3-1: An overview of three different EA approaches

In Table 3-1 an overview of the different approaches is given and in the next three sections they will be described. To give a good illustration of the different approaches the Enterprise Architecture Pyramid of BPTrends will be introduced. This pyramid illustrates how all of the different elements are related within an organization.

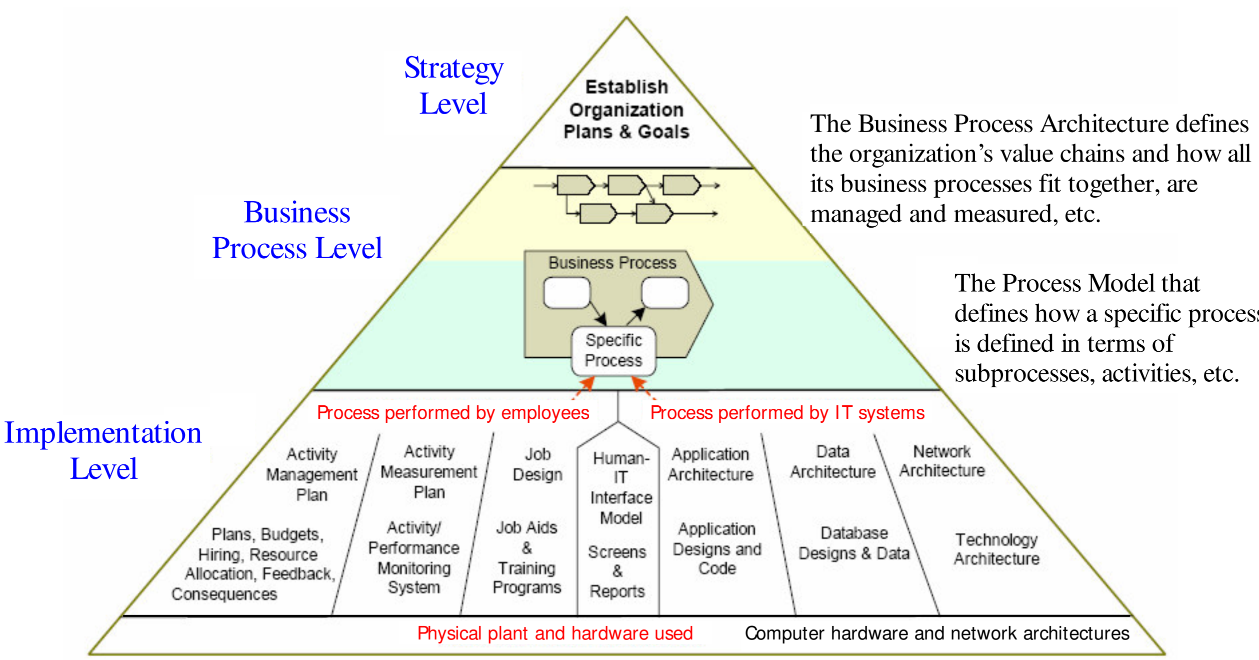
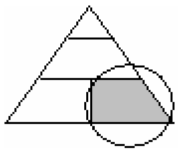


Figure 3-3: The BPTrends Enterprise Architecture Pyramid [BPT2004]

3.3.1 IT-centric



The first approach that will be considered is the IT-centric approach. The objective is to create Business - IT alignment and EA is mainly used for effectiveness and efficiency. With EA IT can rearranged in such a way that it achieve its goals or that it is more efficient and therefore it is cost saving. The figure on the left, which belongs to IT-centric EA, is the pyramid of Figure 3-3. This figure shows that the focus of IT-centric EA lies in the bottom right corner, which is the implementation level of the IT systems. Enterprises, which are using an IT-centric EA, will only consider other areas of the enterprise if that is necessary to define the IT. Business architecture in this context refers to models of the business, which are usually its processes and data models, that support the design of IT-solutions [BPT2004], [Proforma].

The IT-centric EA is an approach that is mostly used in the IT world and it is used for generating an architecture that provides an overview of all the different IT models and resources and how they relate. In this architecture there is a focus on the IT systems and the architecture. The business is only described to lay down the demand for IT and this demand is considered as given and this is not discussed. IT only influences the business when new technology enables new business processes. The IT-architecture has to link-up to the business processes and the EA is a tool to do that as good as possible. Thus there is a lack of communication between the business and the IT. The initiators of an IT-centric EA are the IT managers and this architecture is mainly used by the IT department. Most enterprises and people, who are using EA, are using it from an IT point of view. That is because a lot of companies still consider EA as the responsibility of the IT department.

The model of an EA in Figure 3-4 consists of the four layers from the IAF framework, which will be discussed in section 4.4. This model can be adopted in an EA with an IT-centric point of view. Within Philips, the IT side introduced EA and they adopt an IT-centric approach and they are using the IAF framework. Therefore the four layers will be discussed here and these are [Schekkerman2003]:

- **Business:** business structures, relations, tasks and activities should be defined to the level of detail for which their performance metrics can be validated and their technology support needs identified;
- **Information[1]:** the information layer consists of flows of information, flows of documentation, the needs of information, the sources of information and information exchange with the world outside. This activity defines key information flows and characteristics within a business area at a level of detail that can be used to access their affinities and properly align them in the overall enterprise architecture. It also provide a description of information movement and security services required from the information-systems and technology infrastructure [KUN];
- **Information-Systems:** level of detail is derived from the business and information layer effort, which defines the required solution structure set, functions, features and standards;
- **Technology Infrastructure:** this layer defines the technology services provided, not how they are implemented.

[1] This layer is sometimes mentioned as the Data layer.

The division in these four layers and their level of detail is enough to develop a holistic EA that leads to the proper alignment, validation and implementation of changing business strategies, tasks and activities, information and technology. If necessary the EA elements can easily be extended on in more detailed domain specific solution architectures [Schekkerman2003].

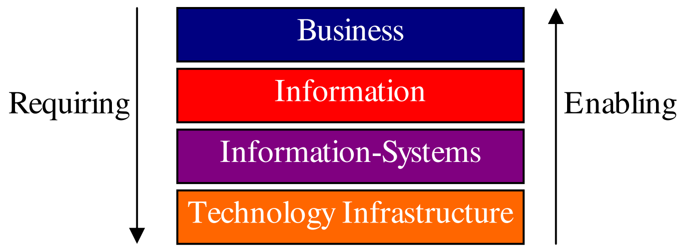
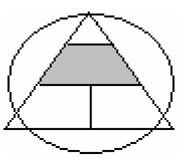


Figure 3-4: The four layers of Enterprise Architecture in an IT-centric approach.

The four layers (see Figure 3-4) each have their own dynamics and they each need their own architectural description. The purpose of this model is to align the business with IT. The model is used for traceability through the layers. That means that with the help of EA, it can be found out which elements from one layer correspondents with elements from another layer. Between the layers it is possible to trace both forward and backward. The arrows in Figure 3-4 show these relations between the layers. The arrow on the left represents the top-down approach. The layer lying on top requires services from the layer beneath it. E.g. to support the process of production an application is required which automatically assign employees to a workplace. The other arrow (on the right) is the bottom-up approach. An underlying layer enables a layer above it to use new possibilities and functions. E.g. with the invention of the Internet a whole new kind of business (E-commerce) arose.

3.3.2 Business process-centric



The second approach that will be considered is the business process-centric approach. The objective of a business process-centric EA is to integrally manage and improve business processes by translating business objectives in actual operations (including IT-support) by a top-down, business-oriented approach. In this way all the processes through the whole organization will be optimally organized in the following way [VanHee2002]:

"… we first design business processes in a more abstract way, without considering implementation, and then we design the information systems and the organization hand in hand."

So first the processes are optimized and then each task in the process will be allocated to an IT system or an employee. The figure in the left top corner, which belongs to business process-centric EA, is the pyramid of Figure 3-3. Although the whole pyramid is in consideration, the focus of business process-centric EA lies in the middle, which is the Business Process Level [BPT2004].

A business process-centric EA provides an overview of all the different business processes (often represented in value chains) and the relations between processes and parties (or their processes) in the context. Processes are redesigned to use both human and IT capabilities in such a way that clients will be more satisfied, i.e. by better and/ or cheaper products or services. In this architecture there is a focus on the business processes and IT is considered to support these business processes. The business processes are being optimized with the help of EA. In the process of optimizing there is no communication with IT and IT is never considered in the first stage of Business Process Redesign (BPR). The initiators of a business process-centric EA are the business managers and it is created and used by the business departments.

The processes in the enterprise can be divided in three kinds of processes. These processes (known in different names) are [VanHee2002]:

- **Production processes or Primary processes:** produce the products and services of the enterprise. These processes provide profit and deal with the customers;
- **Support processes or Secondary processes:** support the production processes. An important part of these processes concentrates upon maintaining the means of production;
- **Managerial processes or Tertiary processes:** direct and coordinate the production processes and the support processes. Objectives and preconditions are formulated and these processes encompass the maintenance of contacts with stakeholders.

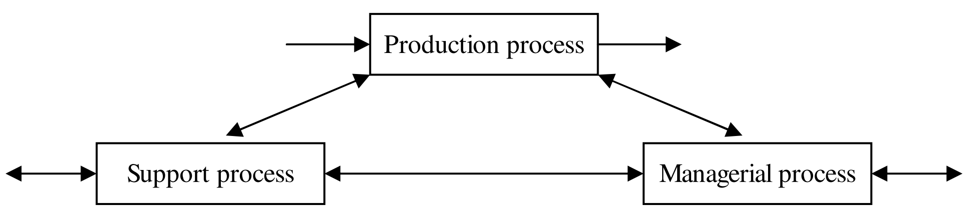


Figure 3-5: The three kinds of processes [VanHee2002]

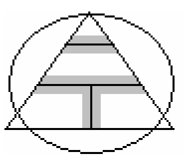
Figure 3-5 represents the three kinds of processes with their mutual relationships and the relationships with the world outside. The support and managerial processes are continuous (they may contain discrete subprocesses), but the production processes are usually case driven and therefore discrete. In Table 3-2 all the relationships of each process with other processes and the world outside will be considered [VanHee2002].

	Input	Output
Production process	• Orders • Raw material and components (support process) • Assignments and purchasing budgets (managerial process)	• Products and services • Return resources to support process after use • Report to managerial process
Support process	• Means to buy resources (from managerial process) • Dispose resources which are no longer functioning (managerial process)	• Place resources to the disposal of production process • Report to managerial process
Managerial process	• Objectives and capital • Reports (production process & support process)	• Performance (often in the form of profit)

Table 3-2: The input and output of the three kinds of processes [VanHee2002]

Thus the business process-centric EA focuses on all the processes within the enterprise, to optimize them and to create new opportunities. These opportunities are either strategy generation or revenue generation. Although the focus lies on the processes, the system and technology architectural components will be utilized. But that is more in a secondary role [Proforma].

3.3.3 Governance-centric



The third approach that will be considered is the governance-centric approach. The objective is to understand the business better, so the management is able to govern (both its performance and conformance, see section 3.4) it better. Because enterprises are getting too complex, there are two strategies with EA that can help in governing the enterprise. The first strategy is to focus on improving management and control systems for the total, integrated enterprise.

The other strategy is to divide the enterprise into a network of smaller, better manageable enterprises, which will be called businesses domains, in order to improve business governance. These business domains are each responsible for their own processes and systems. They exchange business value within a set corporate goals and principles. Corporate governance then aims at managing the portfolio of enterprises to realize performance goals and to monitor compliance to external and internal rules. Lower-level decisions and monitoring are the responsibility of the 'domain owners'. They negotiate and contract relationship between their processes and systems and with parties in the context of the enterprise. The business domains create added value by providing products and services, so they individually can be considered as an enterprise. Together, all the business domains build up the enterprise, so the enterprise is actually a super-domain (a network of enterprises). An EA provides an overview of all the different business domains of the enterprise. Per business domain descriptions from different viewpoints can be made, usually the processes and business data are the most important ones for IT-purposes. The domain owners are responsible for their IT-demand. So it is necessary that there is knowledge of IT in the business domains and not only in the 'functional' IT department [Rietveld et al. 2004 ITG 1].

The figure at the previous page, which belongs to governance-centric EA, is the pyramid of Figure 3-3. Although the whole pyramid is in consideration, the focus of governance-centric EA lies between the different levels (Strategy, Business process and Implementation) and between the employees (organization of the enterprise) and the IT systems.

In Governance-centric EA there is a focus on the collaboration between all the divisions and managers of the enterprise and the focus lies on the communication between Business and IT. The initiators of a governance-centric EA are all the managers within the enterprise and it is created and used by all the departments.

Figure 3-6 shows that in a governance-centric EA the ICT supports the link between the governance and the business. The business has to realise a Business Architecture (BA), which is a combination of organizational visions, business processes and information. The BA defines the starting points for the system developers. BA can be governance-centric when focused on the responsibilities. This definition for a governance-centric BA was described in section 3.1.2.

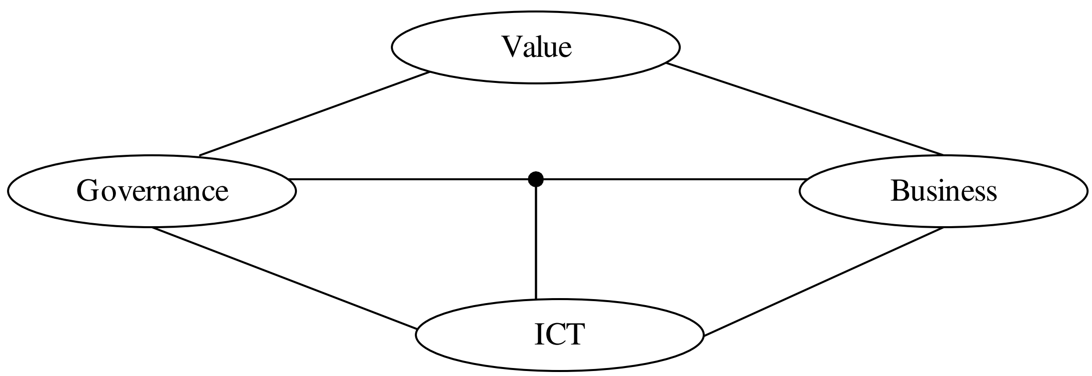


Figure 3-6: ICT supports the link between the governance and the business

The BA brings a systematic process to understand how an enterprise brings value to customers while optimising performance [OCP]. A possible framework of a governance-centric BA is presented in Figure 3-7. This figure will be used to illustrate all the different elements that are involved in a governance-centric approach of BA. These elements are [Hoogervorst]:

- **Mission:** the mission is based on the satisfaction of the customer. It defines the right to exist of the enterprise;
- **Strategy:** the strategy defines how the enterprise will succeed in reaching its objectives. This includes the possibility of organizational learning, which enables adaptation and renewal;
- **Market:** here it is defined how the market should be explored and exploited;
- **Competitors:** here it is defined what the relative position is against competitors;
- **Products and Services:** here it is defined how products and services should be designed;

- **Key Resources:** expresses principles about technology and human resources used in the deployment of products and services;
- **Operating method:** expresses the central financial business focus. The principles define how financial gain is obtained;
- **Customers:** the recipients of the products and services of the enterprise;
- **Environment:** the activities will have an impact on the environment. Therefore principles must be formulated in such a way that it articulates that activities have to be executed in view of the environment;
- **Stakeholders:** the people who are affected or involved by the activities of the enterprise.

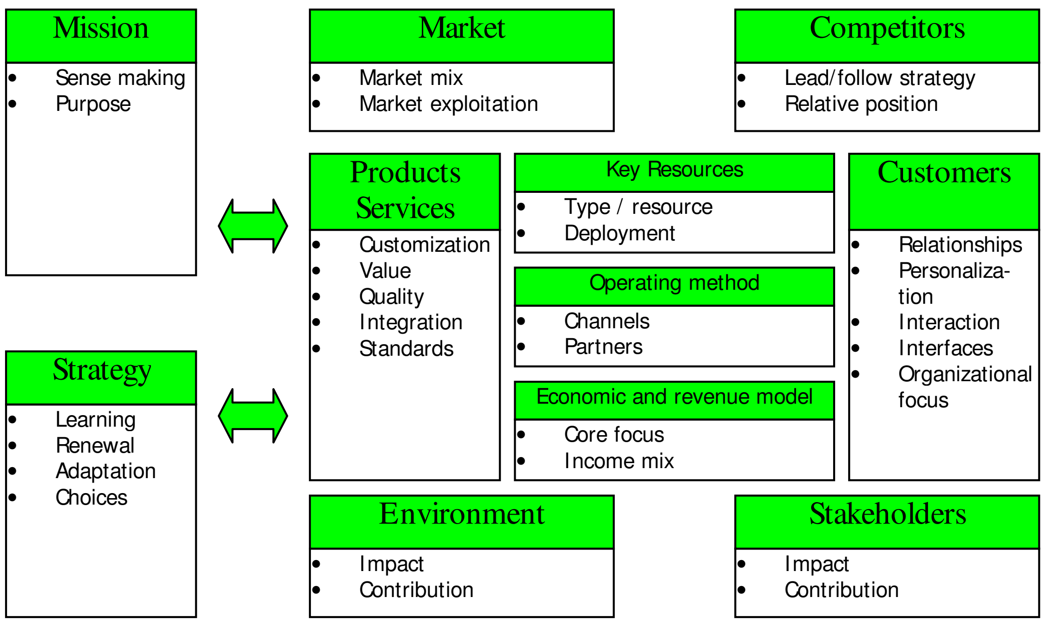


Figure 3-7: A possible Business Architecture framework for a governance-centric approach [Hoogervorst]

Figure 3-7 shows all the different elements that reflect the enterprise and how all these elements are related to each other. This framework is used for monitoring conformance and performance in order to make the right decisions. Conformance and performance will be explained in the next section, which is about governance. In a governance-centric EA the focus is on collaboration and this framework helps to achieve the mission and strategy that are agreed.

3.4 Enterprise Architecture and IT-governance

Enterprises usually start EA to increase understanding, which should then result in better decisions, especially related to IT. Governance is about decision-making and especially about the way how decisions are made in the enterprise. With the help of EA, decisions can be made based on the three different approaches. In an IT-centric approach the governance that is considered is mainly IT governance. Decisions based on an IT-centric EA will only reflecting the IT. But in a Business process-centric approach these decisions are mainly taken in the business and the governance is then part of Enterprise governance. In the Governance-centric approach the EA is used as a mean to govern the enterprise, so IT-governance is an integral part of the Enterprise governance.

Governance is the subject that reflects the questions as: who cares or who decides. Governance is:

Governance: *"the audit, act or manner of conducting the policy and affairs of an organization, and the control or influence of people by constituting rules, standards or principles."*

Governance creates boundaries in which operational management has freedom of acting. Governance sets the rules, standards, and principles that the operational management has to obey and it is a way to control or influence the people in the enterprise.

The definition of IT governance that is used by the IT Governance Institute is:

"IT governance is the responsibility of the board of directors and executive management. It is an integral part of enterprise governance and consists of the leadership and organizational structures and processes that ensure that the organization's IT[2] *sustains and extends the organization's strategies and objectives."*

These definitions show that IT governance is entirely a part of Enterprise governance and it is a high level guidance and control of the development and management of information systems. The main goals of IT governance are in general [Van Hee]:

- Business - IT alignment;
- Minimizing cost of ownership.

Therefore EA might be a way to govern the enterprise better.

IT governance focuses on two subjects: performance and conformance. Performance is how well IT-systems work and what they contribute to an enterprise. Conformance takes things like law, company rules and control on spending into account.

3.5 The benefits of using an Enterprise Architecture

In section 3.1.1 a new definition of EA was given, but why would an enterprise want to develop and maintain an EA? There are several benefits for an enterprise to use an EA and they can be summarized in three words: Better, Faster and Cheaper. But the downside is that an enterprise has to invest in time, money and the underlying organizational and cultural structures to support an EA. The benefits have to exceed the investments to make the EA successful. In this section a list of benefits will be given. Some benefits were already mentioned in previous sections and some benefits are dependent on the approach of EA. The governance from the previous section is also important in realizing the benefits from this section. A lot of these benefits can be realized after understanding what happens in the enterprise. But after that the right decisions have to be taken to change the current situation.

[2] Here, "IT" is understood to encompass the infrastructure as well as the capabilities and organization that establish and support it.

Only after changing the current situation a lot of the presented benefits could be realized [Ambler et al.].

The benefits that will be discussed in the next sections are [AWG1998], [Cap], [Van Hee]:

- Exploiting scale effects;
- Analyzing the current and future state;
- Portfolio Management;
- Communication with the stakeholders;
- Business - IT alignment;
- Customer focused organising;
- Learning;
- Transparency;
- Sensitivity for the context;
- Adaptivity of the organization;
- Entrepreneurial autonomy;
- Clear business rules and conformance agreements;
- Other benefits.

These benefits are sometimes benefits that belong to one of the three approaches, but there are also benefits belonging to all the different approaches. In the next four sections the benefits will be described considering each approaches. The first section describes the benefits that reflect all the approaches.

3.5.1 General Benefits

In this section the general benefits of an EA will be discussed. By using any of the three approaches these benefits can be occur.

Exploiting scale effects

The lack of economy of scale results in avoidable costs. But how is it possible to get a reduction of cost of ownership by exploiting the economy of scale effects? EA is the solution for that. By creating an As-is or a To-be architecture synergy opportunities and redundancy can be found. For instance, in an IT-centric approach the scale effects can be found in synergy between the different applications. In a business process-centric approach the synergy can be found between the different processes. Multiple parts of the enterprise doing more or less the same may adopt the same 'standard' reengineered process and jointly finance new information systems supporting those processes.

Finding synergy opportunities and redundancy is the purpose of the 'towards one Philips' approach. The purpose of this approach is to stimulate collaboration between the different PD's of Philips. With the help of EA it is easier to create a combined architecture that makes it possible to share the same applications. Reducing the cost of ownership means [Van Hee]:

- Sharing of knowledge by using the same architectural standards and principles;
- Exploiting the economy of scale by using shared services.

Analyzing the current and future state
After a blueprint of the current situation is made, it is possible to analyze that. With e.g. the help of simulation, problems or inefficiencies can be identified. When there are problems with the current situation a To-be architecture can be made. Examples of possible problems are: there is a gap between the business and IT in an IT-centric approach, there are new goals or there are new possibilities (e.g. new technology). In this phase an enterprise can for instance execute BPR in a business process-centric approach [AWG1998], [CAP].

The To-be architecture can also be simulated and analyzed. That makes performance evaluation, and consistency checks possible before the changes are introduced. Therefore EA contributes to risk mitigation within the enterprise. Another benefit of the use of EA is that the As-is architecture and the To-be architecture are evaluated and compared in a constant manner.

Portfolio Management of change projects
The next step is to transform from the current situation to the desired one, which is the transformation process. The management of these transformation processes is called portfolio management of change projects. It contains things like migration planning and system integration. The EA specifies the guidelines for the development by its principles and these guidelines are translated to the different design principles [Cap].

Traceability of Arguments
An important objective to use an EA is the 'traceability of arguments'. This has the purpose to make it clear and understandable why a certain decision is taken and to retrieve the arguments that can be used for learning and auditing purposes in a later stage. EA has also a function of control and it brings possibilities like audit and diagnostics. This functionality of EA is becoming more and more important. That is because of the enacting of some laws about audit and diagnostic in the USA; see section 4.2.1 and 4.2.2 [Cap].

Communication with Stakeholders
"EA provides a mechanism that enables communication about the essential elements and functioning of the enterprise." [Schekkerman2003]

This is also a benefit of EA. An Architectural Language is needed to understand each other and EA fulfils in the need of such a language by realizing a formalism to communicate. It makes communication about different aspects of the enterprise and between its different domains possible. In this way stakeholders can understand each other's concerns better. So EA creates transparency for all the stakeholders of the enterprise. The communication does not have to be between all stakeholders in the enterprise. Shareholders for instance would not be interested in all kinds of models about the enterprise. But for other stakeholders EA is the tool to exchange information in a structured way. Examples of a better communication with the help of EA are shared business processes, or the communication between the managers in a governance-centric approach. EA also helps with the external integration with partners. In this way they are able to connect very easily with the systems of the enterprise [AWG1998], [CAP].

Other benefits
Some attendant benefits of using an EA are [AWG1998]:

- When an enterprise has already one or more AD's, like models, the creation of an EA is the right way to handle those AD's in a structured way. In this way EA is the tool for managing AD's. The relationships between the AD's generate an added value, which does not occur in a single AD;
- The AD's can be used by development or maintenance of systems, locations, etc, so the systems are better adjusted to the current situation. In this situation an EA contains also specifications e.g. for a group of systems sharing a common set of features;
- Asset management: the enterprise exactly knows what it owns or uses e.g. how many systems, licences, employees, strategies etc it has;
- Architectures on a divisional level support and get integrated with architectures on a corporate level.

3.5.2 Benefits of an IT-centric approach

In section 3.3.1, the most important objective of an IT-centric approach is Business - IT alignment and that means [Van Hee]:

- Maximal support of the business processes;
- Maximal support of interactions of the business processes (internal and external).

The definition of EA shows that an enterprise has to be described with all its different architectures and the relationships between them. In this way the Business Architecture (including the business processes) can be aligned with the System Architecture, Application Architecture, Software Architecture or Information Architecture. This makes it clear which parts of the business have influence or will be influenced by the IT and in which way. The creation of the EA that represents the current situation is the As-is architecture. With the help of the As-is architecture the need of the business comes clear and this makes it easier to define and select the right components at the IT side [Cap], [Van Hee].

3.5.3 Benefits of a Business process-centric approach

In section 3.3.2, the most important objective of a Business-process centric approach is to manage and improve business processes. A benefit of an business process-centric EA is that it helps in a customer focused organising of the enterprise. Business processes are managed and improved in such a way that the input of the customer is the steering wheel of the business processes. Another benefit of EA is that learning is possible in a structured way by creating feedback loops on every level in the enterprise. In this way, EA is used to fulfil in the needs and demands of the customer.

3.5.4 Benefits of a Governance-centric approach

In section 3.3.3, the most important objective of a Governance-centric EA is a better understanding of the business what makes a better governance of it possible.
By deconstructing the enterprise business model it is possible to assess business results per business domain (instead of functional contributions to the whole) and compare them with relevant benchmarks.

The implementation of EA as a management tool has as benefit that sensitivity for the context is organized on every level of the enterprise. This increases the adaptivity of the organisation, i.e. organisations are able to respond more adequately to changes in their environment. With the help of EA, entrepreneurial autonomy and initiative can be managed at all levels of the enterprise, which results in exploiting the human creative potential. With the help of EA, business principles and conformance will become clear for all the stakeholders.

3.6 Enterprise Information Systems

There are examples of Business - IT alignment like the Enterprise Information Systems (EIS). SAP is an example of a company with a lot of solutions in the field of Business - IT alignment. A lot of companies cannot survive without Enterprise Resource Planning (ERP) software. Within many large companies, SAP has become the facto standard. SAP is an integrated information and managing system in which business processes can be stored and managed. These processes are captured in modules. The data in the modules are mutual exchanged, what leads to a fully integrated system. Also Philips is using a lot of SAP products [Van Giessel], [Wikipedia].

ERP systems like SAP's R/ 3 are organized around the basic economic rationale of the enterprise value chain. It is a process view of a business that regards all transactions as steps in a value accumulation sequence which provides a firm's customers with a final portfolio of desirable product attributes. This movement towards adaptable representations of enterprise value chains would also predict the increasing emergence of business object frameworks where the semantic distance between an economic reality (an enterprise) and the computer representation of that reality decreases even more dramatically. The top-end ERP systems such as those offered by SAP reside in this category where their enterprise orientation aims to integrate accounting transaction processing with workflow, design, and engineering management. ERP systems bring a heavy price in terms of complexity however. They are monolithic and inflexible; the typical business that commits to ERP is committing itself to a process and organizational vision that may not quite fit its reality exactly [OOPSLA'96].

SAP requires that the business processes are adjusted to the SAP implementation. Changes to the SAP implementation can be done, but that is limited and expensive. Therefore a lot of companies have to change their business processes when they want to use a SAP implementation. To do this the business processes have to be described first and after that the proposed changes have to be examined. The Business - IT alignment of SAP is only realized in standard situations. In other situations the implementation or the business processes have to be changed. In the past few years a lot of ERP systems had not always led to the expected results. The reason of this is that after the selection of an ERP system, immediately was started with the implementation phase. This was done without considering how the ERP system would fit into the existing information system architecture. Then it is hard to make the ERP system an integral part of the total information supply.

To prevent these situations the EA tools, which will be discussed in the next chapter, can play an important role. These tools would generate an overview and create an understanding in many situations. That makes it possible to integrate the ERP system into the current environment. With these tools the business processes can be better adjusted to the SAP implementation, or sometimes the changes that are needed to the implementation can be determined. In conclusion EIS systems fulfil in Business - IT alignment in a limited way and EA is necessary to embed these systems into the enterprise [Rijsenbrij et al.].

3.7 A vision at the future

Nobody can predict the future, but it in this chapter some possible developments will be discussed. In these developments EA plays an important role.

3.7.1 The CIO-cockpit

"I have a dream that one day …", these famous words are from a speech of Martin Luther King held at the Lincoln Memorial in Washington D.C. at 28 August 1963. With these words he was emphasizing his perspective for a perfect society in the future. The CIO has also a dream. In his dream everything is connected and working together. He is able to see what is happening with his information systems and he is able to react based on the information he is seeing. On a big screen in his office, he is able to follow the transport of data. He sees if there are failures in the transport of the data and he sees where the throughput of data is slowing down. At any time he is able to see which information systems are producing what kind of data and which processes or other information systems are using that data.

The To-be architecture is generated live from the actual situations; EA-models are not just models but real-time descriptions that support real-time decisions. The CIO will also need to monitor progress vs. the To-be architectures. Projects and their progress are visualized using EA-models. Architectural descriptions are thus directly related to the operative processes what makes them inevitable in the future.

3.7.2 The adaptive organization

At the moment traditional industrial organisations, with separate IT systems, are transforming into adaptive organisations. Several authors have described these adaptive organisations as web based organisations supported by IT systems. These enterprises are cooperating with other enterprises as temporal partners. The keywords are worldwide (especially with the help of the internet), transparent (customers demand better information) and virtual (the outsourcing of processes). This results in a society in which IT supports people and enterprises to develop optimally. In the near future the environment of enterprises will change continuously. Enterprises have to adapt on these changes. To provide the required flexibility components are the solution. Architecture is the way to integrate these new components and legacy systems and align them with existing processes and the ability of the internet [Rijsenbrij et al.].

In Figure 3-8 the transformation process is drawn. To go towards an adaptive enterprise costs a lot of effort. This is depicted as a mountain that has to be climbed. During this climb business has to be redefined with considering the opportunities of IT. The employees, in the IT as well as in the business, will get new capabilities and roles. Application development will be changed dramatically because the development has to be quicker. Complexity, in IT as well in the business, will increase. EA is the tool that will help in the journey to the valley behind the mountain [Rijsenbrij et al.].

When information systems consist of components that are connected to each other, the data is exchangeable. Perhaps the dream of the CIO will be reality then. But to realize his dream a lot of work has to be done. Adopting an EA can be the first step in that.

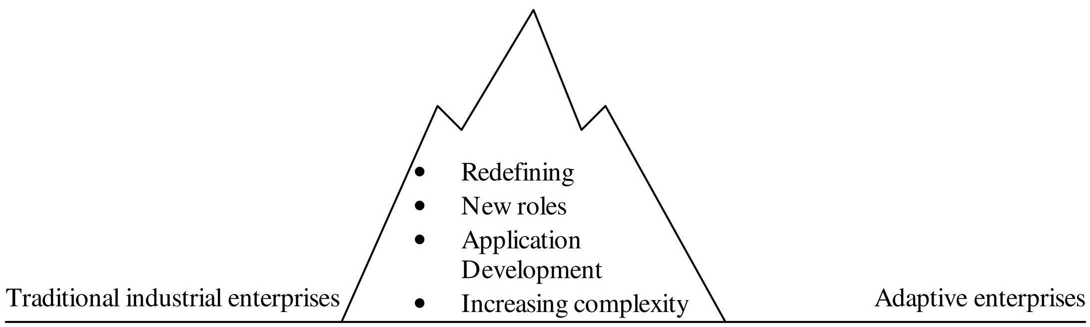


Figure 3-8: the transformation from traditional enterprises towards adaptive enterprises [Rijsenbrij et al.]

4 Enterprise Architecture Frameworks

"If you do not know the Enterprise Architecture framework, you have trouble understanding its contents."
[Vliet et al.]

This quotation shows the importance of understanding a framework. In section 2.3 the most relevant developments in Software Architecture were given. The first relevant development was the introduction of architectural frameworks. In this document the term framework will be used when in fact the term EA framework is meant. In the introduction it was stated that the selection of a framework is the third step of establishing an EA. But what is a framework exactly and why is it needed in an EA? In this chapter these questions will be answered and the IAF framework will be described. Before describing the IAF framework, a short description of the Zachman Framework will be given. That was the first attempt to create a blueprint that defines and controls the integration of systems and their components. In this section also the laws, which have had an important influence on the development of frameworks, will be discussed. In the last section an example of a formal description of the IAF framework will be given.

4.1 What is a framework?

"In a large modern enterprise, a rigorously defined framework is necessary to be able to capture a vision of the 'entire organisation' in all its dimensions and complexity. Enterprise Architecture (EA) is a program supported by frameworks, which is able to coordinate the many facets that make up the fundamental essence of an enterprise at a holistic way" [Schekkerman2003].

In general an EA framework should consists of:

- The kind of models and their relationships, which should be available in an EA and the stakeholders, which belong to each model. The modelling technique which could be used in each model can be defined;
- A description of the process of developing and creating an EA. In most frameworks there is a lack on this aspect. Principles are the way to execute this description.

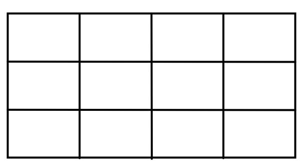
According to these two requirements and the quotation, a framework is a tool to describe enterprise architecture in a structured way. The models and their relationships represent the views of the various stakeholders in a framework. Therefore a framework is a guideline, which helps the architect by making it controllable to establish an EA. The Federal Enterprise Architecture Framework uses the following definition for a framework:

Framework: "*A logical structure for classifying and organizing complex information."*

The definition of an EA framework, which will be used in this document, is derived from this definition. It is specific for EA, thus the 'complex information' is replaced by the 'different architectural descriptions'. Therefore the definition is:

Enterprise Architecture Framework: *"a logic structure for classifying and organizing the different architectural descriptions of the enterprise."*

This definition is a general definition for a framework. It does not specify the models, views, architectures or AD's that are needed in a framework. The different frameworks are different in how they classify the AD's and which views and architectures they take into account. In section 4.3 and section 4.4 two frameworks will be described and then a description will be given of how they classify the AD's. In the definition of an EA, which was introduced in section 3.1.1, every AD identifies one or more stakeholders and aggregates one or more models. In this thesis a framework is defined as a classification of AD's, so every framework identifies stakeholders and aggregates models. The relationships between the models are also part of the framework and this is mentioned in the definition with: '*a logic structure*'. The second bullet of the remarks about a framework, which was given at the previous page, is not part of the presented definition. Here it is considered that a framework can have several processes of developing and creating an EA. It is up to the architect to select the right process for the enterprise. Such process may be called a roadmap and in section 4.4.4 these roadmaps for the IAF framework will be discussed.



The framework supports visualization of:
1. Which architecture products are defined, which are available within the enterprise;
2. The process (to be) followed to produce there architecture products;
3. Who does what;
4. Who decides / governance.

Figure 4-1: The four steps, which are possible with a framework

Figure 4-1 shows a visual representation of a framework together with the four steps that a framework can provide. First the framework is used to categorize the models or principles. This helps to order the architectural products in a structured way and makes it possible to see which products are missing. Then with a framework it is possible to describe, analyse and communicate which process is following another one to produce the models and principles. Thus after one section is filled in, it is possible to recognize the next section that is going to be filled in. In this way, the followed process is known in the future. Then in the next step with a framework it is possible to identify who does what. In this way it is recognized who fills in what section of the framework. Finally with a framework it is possible to identify who makes the decisions in the different sections, so to identify who is responsible for what.

In this thesis the following benefits of frameworks, which are based on the four steps of Figure 4-1 are considered [Cap], [Rietveld et al. 2004 ITG 6], [Schekkerman2003]:
- It aligns various levels / aspects of the enterprise architecture;
- It organizes cross-function traceability of arguments;
- It is a communication model for developing an EA;
- It provides a generic problem space and a common vocabulary;
- It provides guidance on a broader notion of architecture than just what can be conveyed in block diagrams;
- It defines a 'complete' architecture (for instance for a new product);

- It supports the assessment of an existing or new application;
- It evaluates the possible impact of new technology.

Basically a framework puts the models of the enterprise in the right place, by taking the view, stakeholders, etc. into account. An enterprise decides which views, architectures and stakeholders occur in their EA. Based on this information they can decide which framework they should adopt, or to develop their own framework. Section 4.4 will show which of these elements are part of the IAF framework and it will illustrate how the IAF framework is build up.

The problem with many frameworks is that the real environment of the enterprise is much more complex then the environment that most of the current frameworks were designed to address. Most of these frameworks are focused on providing detailed modelling of systems and activities that are only moderately relevant at the enterprise level. Therefore the frameworks are only useful to make blueprints of the current (As-is architecture) and the desired (To-be architecture) situation. The current frameworks aim at Business - IT alignment, but other important aspects of an EA like governance are left out. Therefore these frameworks do not support important benefits of EA like traceability of arguments or performance measurement. In the future these will be crucial objectives of an EA for the enterprise.

4.2 Legalisation

Two laws from the USA, which had an enormous impact on the development of EA and EA frameworks, will be discussed in this chapter. Besides these two laws from the USA the situation in Europe will also be considered.

4.2.1 The Clinger-Cohen Act

The government in the USA played an important role in the development of frameworks is by passing to the Clinger-Cohen Act (sometimes known as the Information Technology Management Reform Act) of 1996. As a result of software application development failures, the Congress wanted to oblige government agencies to develop an EA. They concluded that this was the only way for them to coordinate and manage their IT development efforts, both to achieve government mandates and to assure efficient software development. The Clinger-Cohen Act directs the Chief Information Officer (CIO) of a major US Department or Agency to develop, maintain, and facilitate the implementation of IT architectures as a means of integrating agency goals and business processes with IT. Therefore the frameworks based on the requirements of this act are very IT-centric and pay less attention to the business processes. The introduction of this act has led to a tremendous boost in the use of EA toolsets. The big advantage is that in decision-making not only to the results at short notice are taken into account, but that decision makers create an overall view.

Other important aspects are: is the IT-portfolio balanced concerning risks, technology, ROI, etc. Under the Clinger-Cohen Act US Departments and Agencies are forced to take (well)-considered decisions and these decisions must be traceable. EA in combination with frameworks is a way to support that decision-making process and therefore a lot of US Departments and Agencies started using them [BPT2004], [USGAO], [Kersten et al.].

4.2.2 The Sarbanes-Oxley Act

The Clinger-Cohen Act only applies to US Departments and Agencies. Since January 2002 there also exists an act for all companies that are listed on an exchange in the United States, including non-US companies. This act is the Sarbanes-Oxley Act and it is a direct result of the corporate scandals in the US such as happened at WorldCom and Enron. The Sarbanes-Oxley Act has the goal to restore public faith into the capital markets of the USA. It also requires that the management of the enterprises declare and record the responsibilities about the management and control of the enterprise and the underlying processes. A way to do this is creating an architecture solution with the EA frameworks. A lot of enterprises have decided to use EA and frameworks to fulfil the Sarbanes-Oxley Act. Therefore this Act has an important influence in the increasing use of EA [CapGemini].

4.2.3 Relevant legislation in Europe

Not only in the USA, but also in Europe there were some scandals, e.g. Parmalat and Ahold. Therefore also European governments are working at the design and the implementation of codes and guidelines for good management. The European Commission has introduced plans for a 'Corporate Governance Code' and in the Netherlands the concept code of the commission Tabaksblat has appeared in July 2003. Also the United Kingdom, Germany, France and other countries in the world are revising their codes and guidelines for good management on various points [CapGemini].

As from January 2005 is it obligatory for every enterprise in the European Union (EU) that is listed on the stock exchange, to apply to the International Financial Reporting Standards (IFRS). These international standards aim to improve the relevance, the mutual comparability and so the value of use of the financial reporting. But up to now the EU does not have any laws or regulations to stimulate or enforce development of an EA in their own agencies like in the USA. Currently the initiatives for the development of frameworks mainly originate from consultancy companies like Cap Gemini Ernst & Young. The EU is supporting and participating in several initiatives and projects, like the InfoCitizen[3] project and the EU-PUBLI.COM[4] project, but they are not responsible for the results of these projects [Ernst & Young], [Sckekkerman2003].

4.3 The Zachman Framework for Enterprise Architecture

The first frameworks were designed in the eighties of the last century. Back than IT focused on capturing business requirements as accurately as possible. John Zachman was the first to develop an aid to provide in this need. He identified the need for a blueprint to define and control the integration of systems and their components in the mid 1980s. In 1987 he introduced his 'Framework for Information Systems Architecture', what it was called back then. In his first attempt to create a framework he used three columns: Data, Function and Network. Later he extended his framework with three more columns (People, Time and Motivation) and changed the name into 'Framework for Enterprise Architecture', also better known as the Zachman Framework (see Figure 4-2). This Framework was a breakthrough in

[3] See www.eurice.de/ infocitizen
[4] See www.eu-publi.com or http:/ / research.unisoft.gr/ Eupublicomsite/ home_page/ home.htm

the world and it is still the most used framework in the world [BPT2004], [Rietveld et al. 2004 ITG 6], [Zachman1987].

The Zachman Framework for Enterprise Architecture draws upon the discipline of classical architecture/ construction and engineering/ construction to establish a common vocabulary and a set of perspectives for defining and describing complex enterprise systems. It is a logical structure for classifying and organizing those elements of an enterprise that are significant to both the management of the enterprise and the development of its information systems. Therefore it is possible to identify and describe the planned and existing component parts of an entity (and the relationships between those parts) before the entity begins the costly and time-consuming efforts associated with developing or transforming itself. Since its first publication in 1987, the Zachman Framework has evolved and become the model around which major organisations worldwide view and communicate their enterprise information infrastructure [USGAO], [Zachman].

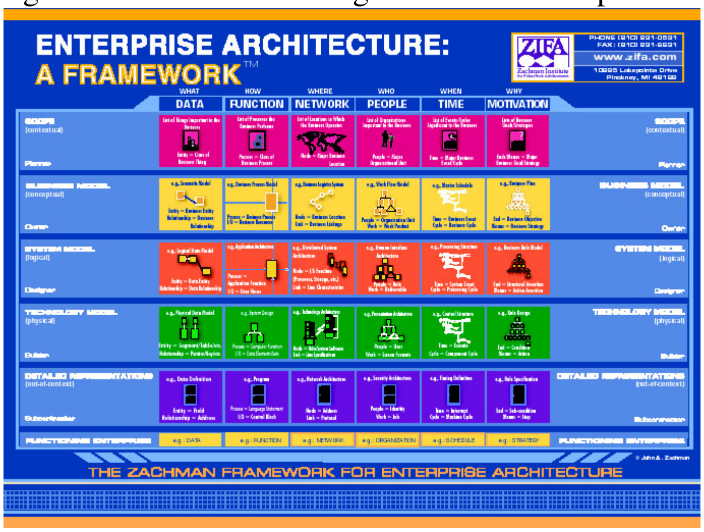


Figure 4-2: The Zachman Framework [ZIFA]

The Zachman Framework consists of six rows (the stakeholders) and six columns (the kind of information). Together they generate thirty-six cells and each cell in the Zachman Framework is unique. When all the cells in a row are combined, they together make up a complete description of that view. The rows are each a separate view and these six views are [Zachman], [Zachman1987]:

- **Planner's View:** This view is about what they like the enterprise to be;
- **Owner's View:** The owner has in mind a product that will serve some purpose. The architect transcribes this perception of a product into the owner's perspective;
- **Designer's View:** After the owners view is made the architect translates that representation into a physical product, the designer's perspective;
- **Builder's View:** After the designer view, the builder applies the constraints of the laws of nature and available technology to make the product producible, which is the builder's perspective;
- **Integrator's View:** This view deals with Configuration Management and Deployment;
- **User's View:** The ERP packages, e.g. SAP from section 3.6, usually offer this row.

The six columns have each a simple underlying model and the six columns are [Zachman]:

- **Data:** This column describes *what* information needs to be stored;
- **Function:** This column describes *how* the things are processed;
- **Network:** This column describes *where* the operations are done;

- **People:** This column describes *who* (people and organizations) are involved;
- **Time:** This column describes *when* something is done;
- **Motivation:** This column describes *why* something is done.

4.3.1 Conclusion about the Zachman Framework

The advantages of the Zachman Framework, which are considered in this thesis, are [Archimate], [Rietveld et al. 2004 ITG 4]:

- **Simple:** The Zachman framework is easy to understand;
- **Comprehensive:** The Zachman framework addresses the enterprise in its entirety;
- **Neutral:** The Zachman framework is totally independent of tools or modelling methodologies;
- **Good questions:** The questions asked by Zachman are needed to support the business description. Zachman taught IT-developers to ask relevant questions;
- **System of models:** different models that are describing different views are needed to describe the enterprise. Zachman introduced that.

The disadvantages of the Zachman Framework, which are considered in this thesis are [BPT2004], [Rietveld et al. 2004 ITG 4]:

- **Business is not central:** The development of the business is not central to Zachman's conception. Snapshot descriptions of processes and structures etc are used as input for IT-development;
- **No alignment:** The Zachman's Framework does not suggest how resources are aligned to support corporate goals and strategies;
- **Too much IT-centric:** The Zachman Framework represents a reasonably comprehensive list of concerns, generated by someone who began with an IT perspective;
- **Supplies a false sense of security:** The Zachman framework is instrumental. It only describes the enterprise but it does not provide understanding of the enterprise. The false sense of security is that architects may believe that if the framework is filled in completely, this will guarantee aligned systems, which is not true;
- **Independent models:** The purpose of a framework is not to fill in all its cells, but a framework should be a logic structure for classifying and organizing the different architectural descriptions of the enterprise. There should be consistency between the models, not just a bunch of independent models;
- **Descriptive framework:** The problem with the Zachman framework is that it is a descriptive framework and that it does not support analyses to reuse models or identify relationships between the models;
- **No principles:** The Zachman framework does not contain principles. In section 3.1.1 the importance of principles was described to give a direction to the EA and to make relationships between principles and requirements possible.

In this thesis the Zachman framework is considered as an important framework. Not that it satisfies the requirements of today. In spite of the major disadvantages, it is still used in many enterprises. It is a framework where the cells can be filled in, but it does not helps to understand the enterprises. But it is the founder of EA frameworks and it still influences the development of new frameworks. Therefore it has made an important contribution in the development of EA, but now it is the turn to other frameworks to distinguish themselves.

4.4 The Integrated Architecture Framework

The Integrated Architecture Framework (IAF) is developed by Cap Gemini Ernst & Young. Figure 4-3 shows that the Zachman Framework had influenced IAF. The arrows represent the influence of a specific framework on another framework, which was developed later. Thus the IAF framework was also influenced by the Enterprise Architecture Planning (EAP) framework. Dr. Steven H. Spewak has written a book[5] about this framework and he defines it as: *"the process of defining architectures for the use of information in support of the business and the plan for implementing those architectures."* According to Spewak the business mission is the most important driver. After that the data required to satisfy the mission is the most important driver. After that the application, which are build to use that data. The least important driver is the technology to implement the applications. EAP has a data-centric approach (the influence of the data-centric EAP is found in the importance of the data layer of the IAF framework what is called the Information layer, see also section 3.3.1) to architecture planning to provide data quality, data interoperability and sharing, adaptability to changing requirements, access to data and cost containment. This is opposed to the traditional view that applications should be defined before data needs are determined or provided for [FAA1998].

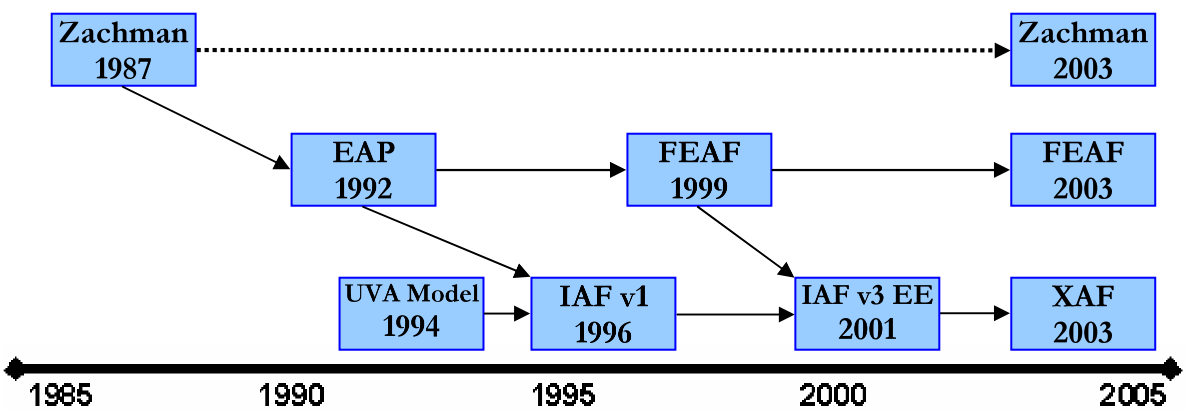


Figure 4-3: The history of the EA Frameworks Zachman and IAF [Schekkerman]

Figure 4-3 shows that existing frameworks influence new ones. The frameworks often have similar ideas and definition, but they differ in focus, scope and intent. Therefore an architect can modify existing frameworks and turn them into a framework that suits his enterprise [Schekkerman 2003].

In Figure 4-1 the four steps that are possible with a framework were given. During this section these steps will be taken into account.

[5] Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology, Steven H. Spewak with Steven C. Hill, John Wiley & Sons, New York City, 1995.

The IAF framework describes how an architecture is built up and it gives insight and overview about the connection between the different aspects areas and abstraction levels. The framework lays the foundations for the communication between all stakeholders about the different aspect areas, abstraction levels and viewpoints. This framework consists of the related aspect areas covering: *Business* (people and process), *Information* (including knowledge), *Information systems* and *Technology infrastructure*, with two specialist areas addressing the *Governance* and *Security* aspects across all these areas. In section 4.4.1 these areas will be described and a definition of each area will be given. Each of these areas is split up in five levels of abstraction: *Contextual*, *Conceptual*, *Logical*, *Physical and Transformational* (see also Figure 4-5). In chapter 4.4.2 they will be described and also a definition of all the five levels of abstraction will be given. There are twenty cells in total and each cell contains a number of documents, schema's, overviews or models [CapGemini2], [Schellen].

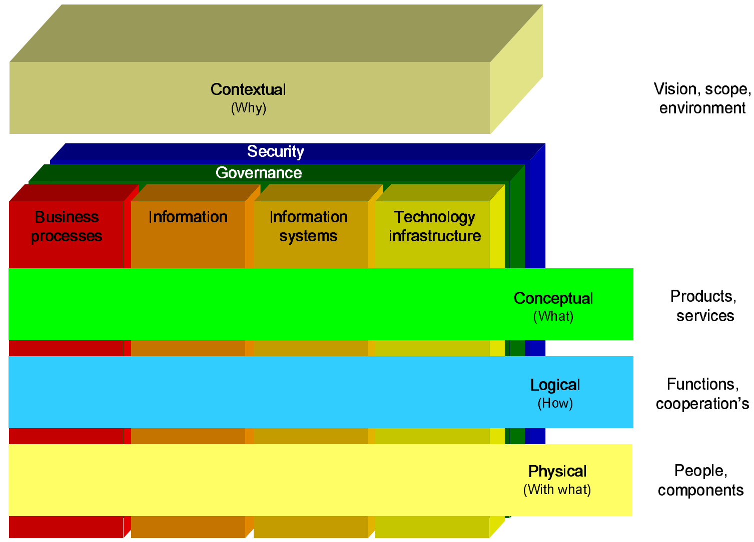


Figure 4-4: The IAF framework [Schellen]

4.4.1 The aspect areas of the IAF framework

The aspect areas, which were described in section 3.3.1, are the columns given in the foreground of Figure 4-4. These four areas have to be related to each other in such a way that a coherent set of relations can be identified. The two specialist areas are integrated across all the aspect areas. In the rest of this section, a definition and an explanation of each area will be given. Together all these areas make up the EA and they define principles, guidelines and rules for [Cap], [Schekkerman2003]:

- The type of components of which the business or system may be composed;
- How these components must fit together;
- How the components communicate and co-operate;
- What assemblies of the components are allowed;
- What functions (communication, control, security and information) the components and component assemblies support;
- How the style expresses the (cultural) values of the stakeholders of that organisation.

Business Architecture: *"the fundamental organization of the enterprise embodied in its domains, their relationships to each other and to the environment and the principles guiding its design and evolution."*

The business is the starting point in developing an EA. It is the actual world, where earning money and doing business is taking place. This architecture contains business structures, relations, tasks and activities that should be defined to the level of detail for which their performance metrics can be validated and their technology support needs identified. The important issues are: the mission, the ambition, the vision, the business strategy, the products and services of the enterprise, the processes that are needed to produce and the organization and the management of the required people and means [KUN], [Schekkerman2003].

Company data, company function, business processes, workflows, stakeholders (including suppliers and customers), supply chains, products, services, business drivers, strategic plans, principles, business goals and objectives, business requirements, business rules, business functions structure and relations, and transformation roadmap play an important role in this architecture [Cap], [Goedvolk], [Greefhorst et al.].

Information Architecture: *"the fundamental organization of the enterprise information assets embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution."*

The Information Architecture is also called the Data Architecture and is about the communication in the enterprise. Data is in this thesis considered as a small unit of information. The Information Architecture consists of a collection of components used to manage valuable enterprise information assets and data management resources. The Information Architecture consists of the flows of information, the flows of documentation, the needs of information, the sources of information, and information exchange with the world outside. Also content management and knowledge management are part of this architecture. This activity defines key information flows and characteristics within a business area at a level of detail that can be used to access their affinities and properly align them in he overall enterprise architecture and provide a description of information movement and security services required from the information-systems and technology infrastructure. The Information Architecture is embodied in components including plans, policies, principles, models, standards, frameworks, technologies, organization and processes that will ensures that integrated data delivers business value and aligns business priorities and technology. The explicit expression of information needs, flows and relations, which are extracted from the business, are necessary to identify the functions that can be automated [Bredemeyer], [DMReview2004], [KUN], [Schekkerman2003].

Strategy, structure, facts, rules, information, knowledge, stakeholders (including information provider, information user), functional and non-functional requirements, information relations, information processes, information objects & relations, information interaction, information flow characteristics, information resources, information locations and information systems roadmap play an important role in this architecture [Cap], [Goedvolk], [Greefhorst et al.].

Information-Systems Architecture: *"the fundamental organization of the information-systems embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution."*

The Information-Systems architecture is also called the application architecture. The level of detail is derived from the business and information architecture effort, which defines the required solution structure set, functions, features and standards. Those are the components of the information-system. The information-systems are the automated services that support specific functions. The Information-Systems architecture describes the structure of the provision of information (independent of organization and technology) and defines the major kinds of information-systems needed to manage the data and support business processes. This includes identifying how the information-systems interact with each other, how they will interact with other business integration elements and how the application and data will be distributed throughout the organization. This is the relationship between each other and the environment from the definition. The purpose is to support every business goal [IEAC], [ISES], [KUN], [Schekkerman2003], [SPC].

IT data, IT function, IT workflow, development, deployment, runtime, roles, Information-Systems services, Information-Systems functions, Electronic Data, Responsibility of Information-Systems, Application portfolio, Information-Systems behaviour, Implementation roadmap, and security impact play an important role in this architecture [Cap], [Goedvolk], [Greefhorst et al.].

Technology Infrastructure Architecture: *"the fundamental organization of the physical IT infrastructure embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution."*

The information-systems are not independent from each other because that is not the most efficient solution. Therefore the common elements are part of the technology infrastructure. This is the foundation on which the provision of information is built and it is also the binding element between all the information-systems. The Technology Infrastructure Architecture describes and identifies the physical IT infrastructure including, the functional characteristics, capabilities, and interconnections of the hardware, software, and communications, including networks, protocols, and nodes. These are the components and the relationships of this architecture. The technology infrastructure is the supporting environment of the information-systems [CDC], [KUN], [Schekkerman2003].

Network, platform, middleware, human/ computer interfaces, control, processing, transmission, storage capabilities, technology infrastructure functions, responsibility of Technology Infrastructure, Technology Infrastructure portfolio, Technology Infrastructure principles, quality of services, technology standards, infrastructure profile, hardware systems profile, communication profile, security profile, technology overview, and transformation plan play an important role in this architecture [Cap], [Goedvolk], [Greefhorst et al.].

Figure 4-5: The IAF framework [Schekkerman]

Governance Architecture: *"the fundamental organization of the authority, management and control in the enterprise embodied in its structures and processes, their relationships to each other and to the environment and the principles guiding its design and evolution."*

In section 3.4 it was mentioned that Governance is about conformance and performance. Performance is about making decisions in the enterprise and this is the management from the definition. These decisions have to be taken in a sensible and structural way, so they can be justified. This is the 'traceability of arguments' from section 3.5.1. The Governance Architecture conforms to the organization, which is necessary to manage and develop the four architectures in mutual adjustment. In this way it oversees the coordination between the Business and the IT and takes care of the alignment between them. Conformance is about audit or control and who is responsible, the authority and control of the definition.

Authority, management and control consist of structures and processes and these are part of the Governance Architecture. They have influence on each other and on the environment. The principles are things like the strategy and the objectives of the enterprise. The authority, management and control have to ensure that the organization sustain and extend its principles. The objective of a Governance Architecture is that it enables sensible, effective and high-quality joint management, under conditions of uncertainty and change, over the period of the relationship [Gray2002], [KUN], [Texas2003].

Security Architecture: *"the fundamental organization of the system embodied in its security components, their relationships to each other and to the environment and the principles guiding its design and evolution."*

The Security Architecture is also an architecture that is integrated in all the four aspect areas. A good security requires an integral approach across all the four architectures and this leads towards a Security Architecture. The definition of a system is the definition from section 2.1 and here only those components are considered that relate to security and how those components mutual relate and interact with their environment. Security Architecture describes how the system is put together to satisfy the security requirements and consider the end-to-end security of IT [KUN], [Texas].

Security profiles, security integration, security plans, and security impact play an important role in this architecture.

4.4.2 The Levels of abstraction

The rows of the IAF framework are the five levels of abstraction and they are also called the levels of concern. The separation of concerns makes it possible to deal with conflict of interest between the concerns. From the top to the bottom of the IAF framework every step is a step more specific in what the organisation needs. The contextual level is about the environment of the problem and the conceptual level is the problem itself. The modelling takes place in the logical level and the physical level. The transformational level contains information of how the enterprise will transform to the situation described in the physical level [GeoNieuws], [Schekkerman2003].

The Contextual Level

The Contextual level describes the relationships with the environment including all the stakeholders and the principles of the architecture to determine shortcomings in the current situation.

This level is the answer to the why-questions. Why does the enterprise need certain processes, a certain strategy or an architecture? The definition contains the following three parts, which are parts of the architecture definition in section 3.1.1 (see also Figure 3-1):

- **Relationships with the environment:** this level describes the parts of the environment, which have impact on the enterprise and the enterprise reacts on the environment with its strategy;
- **Stakeholders:** this level describes all the key stakeholders;
- **Principles:** this level brings the business and other drivers, vision and strategy and their resulting priorities together into a set of principles all of which are described with their implications and priorities. This set of statements is used in the decision making process, providing traceability back to the original business drivers, strategy and vision, and demonstrating the required business-systems alignment;
- **Current situation:** The current situation is described with the As-is architecture, see section 3.2. In the contextual level the As-is architecture will not be made with exception of the introduction of EA. Then there does not exist an As-is architecture, but after that the models will be made in the physical level.

The enterprise decides in this level what it wants in the future regarding to its organisation and environment. Vision, strategy and policies, the risks, critical success factors, the as-is situation, scenarios, and the stakeholders and their interest are important issues in this level [Archimate], [Macauley], [Schekkerman1999].

The Conceptual Level
The Conceptual level describes the missions of the different systems of the enterprise.

This level is the answer to the what-questions. What does the enterprise want and what does it need? The definition contains the following part, which is part of the architecture definition in section 3.1.1 (see also Figure 3-1):
- **Mission:** this level describes the requirements of the enterprise by giving the requirements for every system. These requirements will be described independent of the solution. In section 3.1.1 it was already mentioned that the requirements are called the mission of the system which appears in the description of an architecture;

This level enables the actual design due to the fact that if it is not known what is required, it could never be designed. It details the services and the interactions between these services in support of the principles defined in the Contextual level. This level is about the concepts or functionalities of the enterprise, in other words what the enterprise wants. Requirements, principles, constraints, and services models are important issues in this level [Archimate], [Cap], [Macauley], [Schekkerman1999].

The Logical Level
The Logical level describes the architecture of the ideal logical solutions

This level is the answer to the how-questions. How can the enterprise achieve the best solution? The definition contains the following part:
- **Ideal logical solutions:** this level describes the best solution. It models what actually is required, without taking 'real life' constraints into account. This level describes the models/ views, viewpoints, rationales and concerns from Figure 3-1 in section 3.1.1.

This level describes the best solution that fully reflects the requirements from the Conceptual level. That gives a long-term direction and guidance for short-term decisions. The key decision is to select (with the business) the solution alternative that delivers the services required, in a way that best addresses the guiding principles from the Contextual level. Internal construction, architectural principles, and internal behaviour: cooperation of components are important issues in this level [Archimate], [Cap], [Macauley], [Schekkerman1999], [Schekkerman2003].

The Physical Level

The Physical level describes the architecture of the products and techniques, which can be implemented or bought.

This level is the answer to the with what-questions. With what components is the system we going to be build? The definition contains the following part:

- **Products and solutions, which can be implemented or bought:** this level describes the products and techniques that actually can be realised in certain point in time. It models what actually can be achieved. This level describes the models/ views, viewpoints, and concerns from Figure 3-1 in section 3.1.1.

Very often there will be a gap between the ideal logic solution from the Logical level and the solution that actually can be realised from this level. This will be caused by lack of sufficient support from products and techniques that can be implemented or bought. The gap needs to be closely monitored and closed as new products and solutions come available. The physical level details the design principles, standards and guidelines, including component grouping in critical areas as well as deployment models. It is at this level that solution frameworks and architectures can be used to accelerate development of the physical architecture, improve the quality of the architecture (by using proven solutions) and reduce project risks. The physical solutions are based on change, redesign, products or techniques. Physical scenarios and assignment of roles are important issues in this level [Archimate], [Cap], [Macauley], [Schekkerman1999], [Schekkerman2003].

The Transformational Level

The Transformational level describes the change from the existing situation towards the future situation.

This level is the answer to the question when. When does the enterprise fulfil the transformation process? The definition contains the following part:

- **The change from the existing situation towards the future situation:** this level describes the change and how an enterprise must deal with it. In section 3.2 the transformation process was described.

This level describes the impact of the proposed solutions from the Physical level for the organisation. This level is not always mentioned in articles about the IAF framework. In this thesis this level is considered as important because EA is also about realizing improvements. This level describes how to realize the improvements. Therefore this level is described in this thesis and used in Figure 4-5. Realisation, the As-is situation, implementation, and migration are important issues in this level [Archimate], [Cap], [Schekkerman1999], [Schekkerman2003].

4.4.3 The relationships in the IAF framework

In section 3.1.1 it was mentioned that describing the relationships between the different architectures are the added value of an EA. Therefore the IAF framework must support the relationships between the different models and principles that build up the different AD's.

In the IAF framework the information architecture is the link between the business and the IT. With the IAF framework the actors (who are doing it), the objects (the subjects) and the functions are connected together. In the framework the principles are, from the left to the right, the link between the different architectures. They form the key in finding the relationships between the different architectures. In section 3.1.1 there was an example given of two principles that are related to each other. Now six principles will be given and these are derived from each other and these six principles are from the contextual level of the six aspect areas (architectures) that were discussed. Together they make up the set of principles. The different principles are:

- **Business principle:** *"All employees have to access data, without special software, all over the world."*
- **Information principle:** *"The owner of the data is responsible for the actuality of information."*
- **Application principle:** *"Components must be reused."*
- **Technology-infrastructure principle:** *"Security profiles and business rules will be implemented in the network layer."*
- **Governance principle:** *"The infrastructure will be centralised managed, application support will be decentralised organised."*
- **Security principle:** *"Only owners of the data can change or delete information."*

The governance principles are the general principles, which were mentioned in section 3.1.1. The principles are the method to link the different sections of the IAF framework together. The right approach is to that from the left to the right. From the business principle it is possible to derive the information principle and so on. The IAF framework does not only contain horizontal relationships between the different columns, but also vertical relationships between the different rows in a column. Now the transition between the different levels of abstraction will be described. To go from one level to another, they have to be separated. The IAF framework separates [Cap]:

- **The context from the requirements:** the clear distinction between the context and the requirements is necessary to identify all environmental aspects, stakeholders and elements that can influence the progress of the architecture trajectory. The contextual level is key for the success of the overall result, by identifying the mission, the vision, the business strategy, the business drivers, the principles and all the stakeholders involved;
- **The requirements from the logical solutions:** the clear distinction between requirements (conceptual level) and logical solutions (logical level) helps to separate business drivers, principles and requirements from logical solutions;
- **The logical solutions from the physical solutions:** the clear distinction between logical solutions and physical solutions helps to design the best-fit solution and create boundaries for migration path toward that solution;

- **The physical solutions from the transformation path:** the timeframe to implement a physical solution can differ in time, priority and scope based on technological or organisational constraints. So for each aspect area the impact of the transformation has to be defined and the dependencies has to be clear.

The relationships between the principles are the key in finding relationships between two different sections of the framework. Because all the rows in a column are related to each other it is possible to find the matching principle of a section. Then the matching principle of the other section can be found because the principles are related to each other. Then the related model or models can be found in the target section, because the rows in the column are related to each other. In this way the consequences of a change in one section can be find in every other section.

4.4.4 Roadmap

In section 4.4.1 and 4.4.2 the definitions from all the rows and columns were given. But in what order do the cells have to be filled in? Are there conditions or not? There are a number of approaches to fill in the framework. In this thesis such approach will be called a roadmap and the definition for a roadmap is [Cap]:

Roadmap: *"a process pattern describing how to run an architectural engagement for specific architectural objectives. It specifies architecture content as well as the engagement process."*

A roadmap is a way to fill in the framework in a structured way and it takes relationships between the different cells into account. It is the shared vision (internally and externally) of how the objective can be reached. The benefits of a roadmap are [Cap]:

- It speeds up the process;
- It focuses on what is important;
- It avoids redundant, non applicable elements;
- It identifies existing assets;
- It works only to the level to answer the client's questions;
- It defines the stakeholders who will be needed (internally and externally):
    - Who will participate;
    - What is the required availability;
- It defines / validates the budget and the timeframe of the engagement;
- It guarantees the consistency of the approach.

An enterprise has to select a roadmap that suits their architecture practice best. There are a few approaches but generally speaking there is always gone from the left top corner to the right bottom corner. It is possible to go one cell down or one cell to the right. An example of a common approach is to fill in first the row above (contextual) and then fill in the rest with the 'saw cog' technique. First the conceptual and logical cells from the business are filled in. Then the conceptual cell from the information architecture is filled in. After that the logical level of it is filled in and so on. In all columns at least the questions at the conceptual and logical level must be answered, before starting a change project decisions in the physical level.

A Roadmap can be an initial process or it can emerge. When it is an initial process it follows a fixed pattern, which is determined in advance. But to maintain the framework a special roadmap can emerge. Somewhere in the framework a problem comes up and there is a necessity to redesign it. The influence that it does have on other segments of the framework and in which order these frameworks are addressed is the roadmap that emerge.

A roadmap can also define who does what and who decides what. These two steps (that are possible with a framework) are close connected with governance. But an enterprise can make decisions about these two steps and make it part of the roadmap. The not only the process is recorded but also definitions and executions with the help of the framework.

4.5 Evaluation of the Integrated Architecture Framework

This section contains of three subsections in which the IAF framework will be evaluated. In the first subsection the IAF framework will be compared with the definition of frameworks. The second subsection will contain a formal description of a framework and the last subsection will contain some points of the IAF framework that are used as sales arguments.

4.5.1 Evaluation versus definition in this thesis

In this section the introduced definition of an Enterprise Architecture Framework will be compared with the IAF framework. In section 4.1 a framework was defined as *"a logic structure for classifying and organizing the different Architectural Descriptions of the enterprise."* In section 2.3 an explanation of an AD was given: that an architecture is described by one AD and that an AD contains or reflects principles, models, views, viewpoints, stakeholders and concerns. All these concepts are part of the IAF framework. The principles and models are the products of an EA and they fill the twenty different cells of the IAF framework. In Figure 4-6 these cells are all the cells except the yellow ones.

On the left of Figure 4-6 there are five concerns and that are the rows of the framework. These are the concerns from the AD's, which are mentioned in section 2.3 and the concerns are split up into five levels of abstraction. Every AD identifies in this way more than one concern and the blue and purple cells together deal with one concern.

	Business Architecture 1	Information Architecture 2	Information-Systems Architecture 3	Technology-Infrastructure Architecture 4
Contextual level / Concern 1				
Conceptual level / Concern 2				
Logical level / Concern 3				
Physical level / Concern 4				
Transformational level / Concern 5				
	Business-system		IT-system	

Figure 4-6: The IAF framework

The columns of the framework are the four architectures (see the four layers of section 3.3.1) and this is typical for Business - IT alignment. The enterprise has different stakeholders, and they have different interests. The stakeholders of the enterprise are interested in these different architectures, so the stakeholders influence the different columns. The columns are the different viewpoints of an EA and the different stakeholders of the enterprise have different viewpoints. The business architecture and the information architecture together make up the business-system. The information-systems architecture and the technology-infrastructure architecture together make up the IT-system. The IAF framework divides a business system into an integral viewpoint and an information viewpoint. It divides an IT-system into information systems and an technical infrastructure. Viewed from an integral viewpoint, the business system is the domain of the business architecture. Viewed from the information viewpoint the business system represents usage and control of information and knowledge and it is the domain of the information architecture. There are also other views then those four architectures. These views are created based on other viewpoints or themes. Viewpoints deliver added value to the aspect areas by addressing and focusing on these specific themes, covering all aspect areas and levels. Examples of such views are the governance viewpoint and the security viewpoint [Archimate], [Schekkerman2003].

The cells together viewed from a viewpoint are making up a view. A view belongs to one viewpoint. In Figure 4-6 the red and purple cells belong to the third architecture and that is the information-systems viewpoint. The two special viewpoints (security and governance) are a fixed part of the architecture approach and they cover all the aspect areas in connection, so the views are based on every cell of the framework [KUN].

4.5.2 Test versus formal description

During the study Computer Science, formal languages and automata are a part of the education. These formal languages are used to describe discrete systems. The IAF framework can also be described with such language and that will be done in this section.

A grammar G is defined as a quadruple [Linz]

$$G = (V, T, S, P),$$

Where

- V is a finite set of objects called variables,
- T is a finite set of objects called terminal symbols,
- $S \in V$ is a special symbol called the start variable,
- P is a finite set of productions.

It will be assumed that the sets V and T are non-empty and disjoint.

The production rules are the heart of the grammar; they specify how the grammar transforms one string into another, and through this they define a language associated with the grammar. Here, it will be assumed that all production rules are of the form

$$X \rightarrow Y$$

Let $G = (V, T, S, P)$ be a grammar. Then the set $L(G) = \{w \in T^*: S \overset{*}{\rightarrow} w\}$ is the language generated by G.

If $w \varepsilon L(G)$, then the sequence $S \rightarrow w_1 \rightarrow w_2 \rightarrow \ldots \rightarrow w_n \rightarrow w$ is a derivation of the sentence w. The strings $S, w_1, w_2, \ldots, w_n$, which contain variables as well as terminals, are called sentential forms of the derivation [Linz].

Now a practical example of the IAF framework will be given. In Figure 4-7 a screenshot is taken from the test case with Metis. During the example the information will be taken from the input in Metis. Consider the grammar G = (V, T, S, P) with:

V = {1;1, 1;3, 2;3, 3;1, 3;3, 3;4, 4;1, 4;3, 4;4, 5;1, 5;3}

T = {Strategy, Environmental Factor, Concept, Strategic Requirement, IT Implication, Concept, IT Principle, Design Rule, IT Requirement, Business Function, Role, Application Function, Application Building Block, Application Environment, Business Process, Application, Enabling IT Technology, Application Product, Service, Datastore, Transition Plan Item, Project}

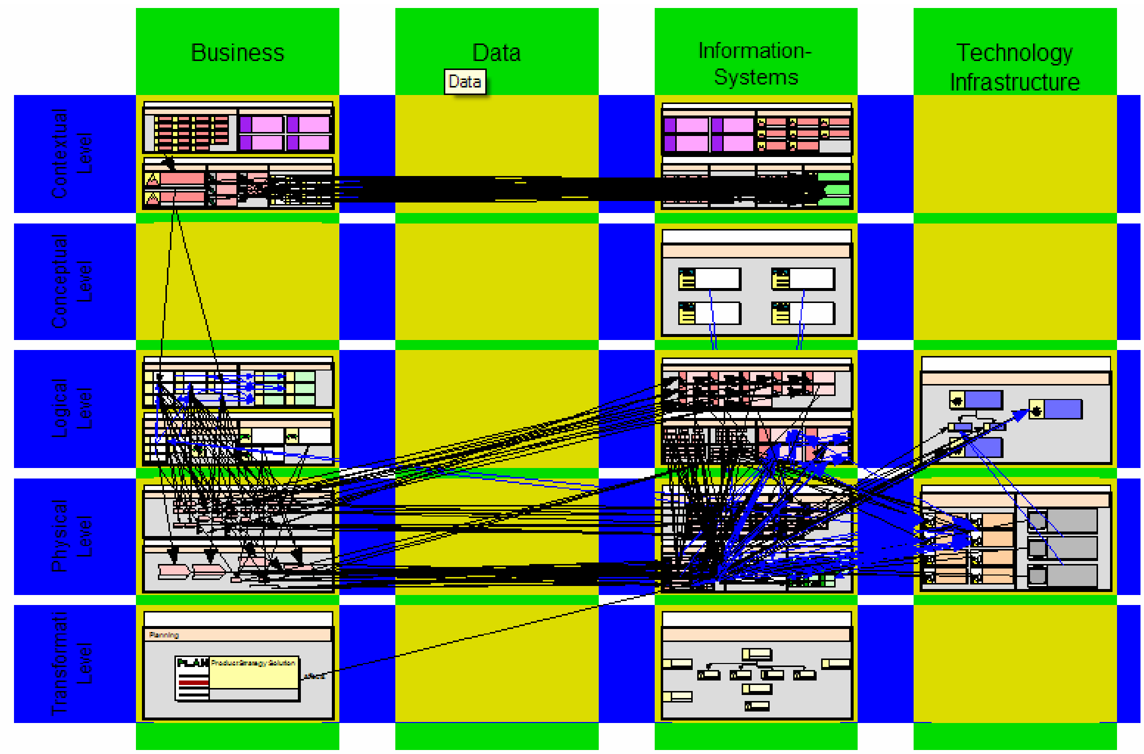


Figure 4-7: A screenshot from the IAF view in Metis

With P given by:

S → 1;1
S → 1;3
S → 2;3
S → 3;1
S → 3;3
S → 3;4

3;1 → Business Function 4;1
3;1 → Business Function 4;4
3;1 → Role
3;3 → Application Function 4;3
3;3 → Application Function
3;3 → Application Building Block 4;3

S → 4;4
S → 5;1
S → 5;3
1;1 → Strategy 1;1
1;1 → Strategy
1;1 → Strategy 1;3
1;1 → Strategy 3;1
1;1 → Environmental Factor
1;1 → Concept
1;1 → Strategic Requirement 1;1
1;1 → Strategic Requirement 1;3
1;3 → Environmental Factor
1;3 → Strategy
1;3 → IT Implication
1;3 → Concept
1;3 → IT Principle 1;3
1;3 → Design Rule
2;3 → IT Requirement 4;3
3;1 → Business Function 3;1
3;3 → Application Building Block 4;4
3;4 → Application Environment
3;4 → Application Environment 3;4
4;1 → Business Process 3;3
4;1 → Business Process 4;3
4;1 → Business Process
4;3 → Application 3;3
4;3 → Application 4;3
4;3 → Application 3;4
4;3 → Application 4;4
4;3 → Application
4;3 → Enabling IT Technology 4;4
4;3 → Application Product
4;4 → Service 3;1
4;4 → Service 5;1
4;4 → Service 3;3
4;4 → Service 4;3
5;1 → Transition Plan Item 4;3
5;3 → Project

In this example the cells of the IAF framework are numbered as follows x;y, where x is the number of the row and y the number of the column. Striking facts are:

- **The number of starting points:** in this language there are nine different starting points. The two cells at the last row are also starting points. It cannot be true that the projects do not support any strategy as well on the business side as on the IT side. With a formal description it is clear what the possibilities are and only predefined languages can appear;
- **The roadmap:** in section 4.4.4 the roadmap was discussed. In that section it was stated that roughly the cells have to be filled from the left to the right and from the top to the bottom. But there are productions like for instance 4;4 → Service 3;1, which is pointing to the left and a row above. Or the objects are put in the wrong cell, or the architectural view is not compliant with the IAF framework.

Principles are a way to deal with the facts above. The IAF framework has no limitations and an enterprise has to define what is allowed and what is not with the help of principles.

Now an example of a path that can be generated with this language will be given. In Figure 4-8 a path is given. This is also a screenshot from Metis, which was generated during the test case what is described in section 7.5. The property on the right of the figure explains the relationships between the objects. The matching language is:

S → 1;1 → Strategy 3;1 → Strategy Business Function 4;1 → Strategy Business Function Business Process 4;3 → Strategy Business Function Business Process Application 4;3 → Strategy Business Function Business Process Application Application Product

With the given language all the existing paths from the IAF implementation, of Figure 4-7, can be created.

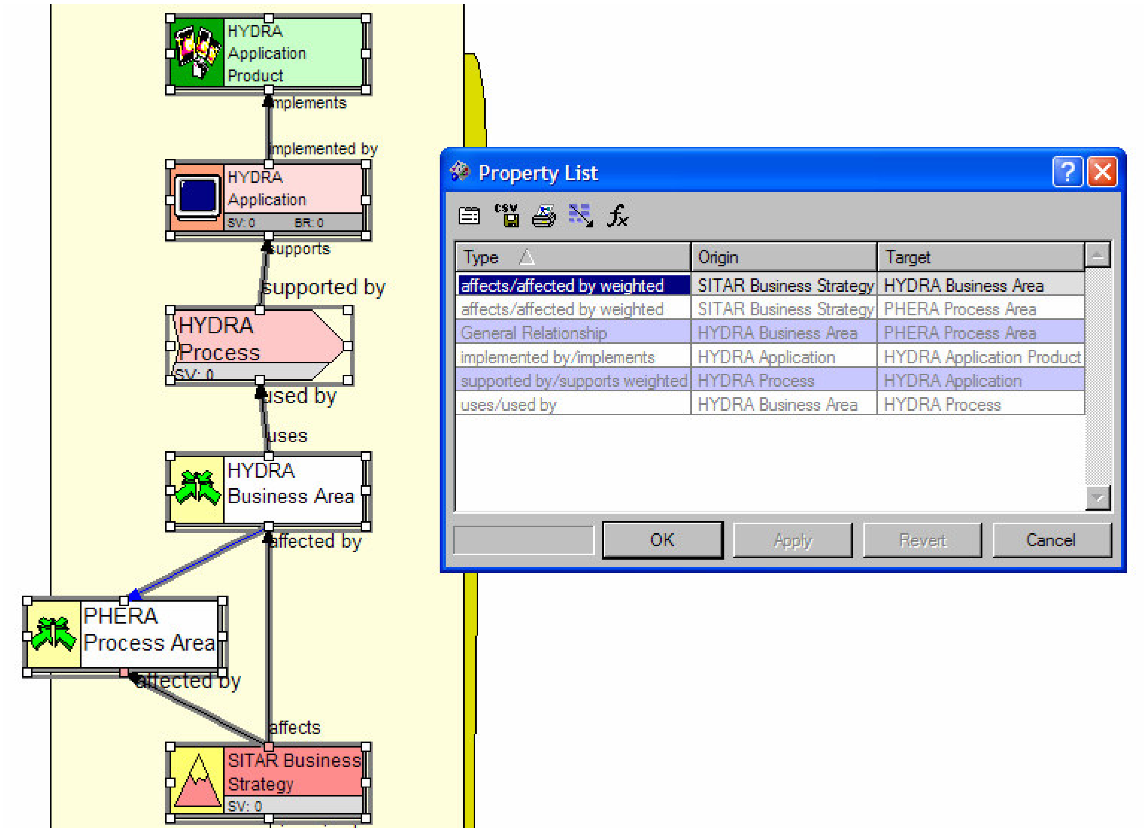


Figure 4-8: A path from the language of the IAF framework

The formal description of the IAF framework has a lot of benefits. It is clear how the framework is built up. All the roadmaps are predefined, thus it is clear what is possible and what not. If certain objects are used in different cells like strategy it is not a problem. It is clearly predefined when a strategy can be connected to a business area and when not. Also this language allows more than is actually possible. For example an IT principle can lead to a Design Rule, but not to a Environmental Factor. These kinds of conditions are not part of the language now. Also it is sometimes forced that some objects always have a target. For instance cell 2;3 has only one production (2;3 → IT Requirement 4;3), which means that an IT Requirement has always a relationship with an object in cell 4;3. In chapter 7 the tool Metis will be considered more closely. This tool uses templates and a template contains predefined objects and relationships between them. The created language in this section could be an ideal starting point for creating an IAF template. The benefits of this formal description are:

- **Starting points:** the starting points are clear;
- **Roadmap:** the roadmap is clear from the starting point and all the relationships that are possible between the different objects;
- **Formal:** it is a formal description that eliminate different assumptions of what belongs in which cell and which object may relate to a certain object;
- **IAF:** this description can describe the IAF framework in such formal way.

4.5.3 Assessment of IAF sales arguments

The IAF framework is focussed on Business - IT alignment, but it also supports communication among all stakeholders about relations between the aspect areas and the levels of abstraction seen from different viewpoints. The topics that could be important for the architecture are on the intersections of these relations. The IAF framework as a communication framework provides sufficient insight and overview about the connexion between the different aspect areas and levels of abstraction. It is a structured, top-down, business oriented approach that allows the development of the technical architecture of the IT for an enterprise or part of an enterprise [CapGemini3].

Selling points about IAF include [Cap], [Schellen]:

- **Business driven:** The lead has to come from the business in order that the technical architecture and applications developed are in line with the business. Technology has a significant part to play but must be tempered by the direction in which the business is going. Thoughts behind the IAF framework are: the business benefits are leading, choices are based on the business, it is business centric, the business principles are mandatory and business scenario thinking, technology is enabled by business benefits. The IAF framework supports Business - IT alignment and it has an IT centric approach. The IT department is the owner of the architectural descriptions and therefore the business is seen from an IT point of view. The architectural descriptions are created and used by the IT department, which is described in section 3.3 and Table 3-1;
- **Collaborative:** To get the most from any integration project, the business works in conjunction with the technical architects who are responsible for designing and implementing the solution. Business is here only used for input, the business is not creating its own architectural description;
- **Rapid:** The approach ensures that deliverables are produced at an early stage. This allows the business to see the ultimate direction relatively quickly and to measure the effectiveness of the approach. This approach allows the pragmatic deployment of the framework in many different scenarios, both by using only the relevant parts of the framework and by supporting iterative working across the streams. This flexibility minimizes the traditional effects of a waterfall approach and ensures coherency across the aspect areas [Macauley].

To conclude an overview will be given about the things what IAF is or is not. The IAF framework is thus a descriptive framework in which AD's can easily ordered. Besides this the principles are a benefit. They make relationships and dependencies between the different AD's traceable. [CAP].

IAF is:

- A framework showing aspect areas;
- A framework showing abstraction levels;
- A framework showing integrated relations with the help of principles.

IAF is not:

- An architecture method;
- A process model;
- A development framework;
- A governance framework.

5 Enterprise Architecture Tools

EA and its frameworks are discussed in the previous chapters. But how is it possible to create and manage the different models in an EA? This can be done by software tools. These tools should be able to create, model, and store the different models and principles. Also they have to make it possible to create relationships between these different models and principles. A tool has to be able to support different type of users like managers and architects. Architects have to be able to collaborate at the same models at the same time. Traditional office tools like Word, Access, Excel and Visio fail due to [Peyret2004]:

- **Multiple types of users:** there are a lot of people who are using an EA, like IT strategists, central IT architects, IT planners, business managers, IT managers, and project architects. Every group needs to represent the same (a representation of an aspect of the enterprise) from different viewpoints;
- **Distributed systems:** the rise of multiple-tier architectures based on Internet technologies has added technical complexity to the management of enterprise architecture. One consequence is that IT operational staff, such as systems managers and database administrators, need to be more involved in EA work;
- **Expansion of the business analyst role:** the business analyst has to be involved in EA because Business Process Modelling (BPM) is an important part of EA. To realise BPM special tools are required, because the embedded modelling tools (tools that are directly provided by the execution platforms) are complex and inflexible;
- **Relationships between the models:** in section 3.1.1 it was described that the relationships between the models are the added value of an EA. But in the traditional office tools it is impossible to generate relations between the different models;
- **Coordinating decentralized architecture teams:** in enterprises with multiple business units each business unit can make its own IT decisions. It is difficult to coordinate the work of the different architects in the different business units;
- **Adapting a common architecture to local considerations:** although architects of different business units collaborate to create a common enterprise architecture, they adapt it to their local geographical considerations and the financial considerations of their business units. So the more complex the corporate structures are, the more complex the EA will be and that makes a common vision for EA more difficult;
- **Business - IT alignment:** the architectures of the enterprise take the business and technology constraints into account. But they lack the methodologies, documents and tools to demonstrate Business - IT alignment systematically. Without a systematic approach, the dependencies and links between the architecture elements and models are not tracked. When business priorities change, architects have difficulties to reassess the IT priorities according to the new business priorities.

Traditional office tools are not sufficient in giving a solution to the problems, which were described above. To offer a solution to these problems, a special kind of tool is needed in which architects can create, manage, and store different models from different viewpoints in a structured way. The different models should be accessible to all the users of an EA and the models should relate to each other. This chapter contains an overview of the special kinds of tools that are available for EA. Also this chapter contains a description of the purposes they serve. First of all an introduction will be given and this is an introduction about the roles that these tools have to fulfil. Section 5.2 contains an overview of the history of EA tools. After that an overview of the different approaches, which are supported by the different EA tools, will be given.

5.1 Introduction to EA tools

The EA tools offer a solution for the problems from the previous section. To fulfil in the needs that were described in the beginning of the previous section, EA tools provide [Peyret2004]:

- Modelling tools
- A repository

The focus of EA becomes more and more holistic, thereby necessitating the use of comprehensive modelling tools to analyse and optimise the portfolio of business strategies, organisational structures, business processes/ tasks and activities, information flows, applications, and technology infrastructure. Different EA tools exist to help architects model, store, manage, and share information about EA and these tools support all the different views that are needed to represent the enterprise [BPTrends].

EA tools provide a graphical interface to model [IFEAD]:

- Business strategy;
- Business activity;
- Related information and data structures;
- Applications to support the business activities and processes, and supporting infrastructures;
- The arguments of how the four models above relate.

Selection of a comprehensive modelling tool requires a focus on more than just functional requirements or price. The tools tend to be more industrial-strength than common diagramming tools like Microsoft Visio and they enable multi-user support. The obtained models and information, which are generated by using such tools, have to be stored into a repository. This is a special kind of database in which not only the models and information can be stored, but also the arguments of how those different models relate to each other. This enables traceability of arguments (see also section 3.5.1) [Schekkerman2003].

A group of people must manage an EA and they will necessarily need to keep track of a wide variety of documents, diagrams, models, and charts. This can be done by an EA tool, which organizes the information and stores it in a repository. With the help of an EA tool it becomes possible for any member of the group to quickly obtain all of the documents, diagrams or models that refer to a given business goal or process, or to a specific activity or application used in a process. This helps in improving collaboration between the business analysts and IT specialists. Current tools are also suitable for non-technical users, e.g. strategic planners can effectively maintain their strategy models; business process owners can maintain their business process models [BPTrends].

5.2 History of EA tools

EA tools exist for more than ten years now. Some tools have evolved from development modelling tools like Popkin or from operation optimizing tools like ASG-Rochade. In 2003 the market of EA tools was worth $250 million and it is still growing with 15% to 20% per year. The big growth in this market is also the result of the introduction of the Sarbanes-Oxley act (see also section 4.2.2), and more and more enterprises expecting that EA can help them comply [Peyret2004].

In the history of the EA tools there are roughly three generations of EA tools. These generations are [Peyret2004]:

1. **First generation:** in terms of collaborative features, a decade ago the most advanced EA tools allowed users to publish and navigate through models with HTML. This enabled different architects (including architects outside the firm) to access the most current architecture models. MEGA International, for example, delivered the first generation of collaborative support in 1995, and Proforma still offers HTML generation functionality;
2. **Second generation:** the second generation of collaboration functionality in EA tools allowed XML generation. Typically, this was deployable on any Web server and provided a better browsing experience of graphical models: For example, XML modelling included a local zoom feature. Popkin Software's SA Information Publisher using SVG offers this collaborative functionality. Computas' Metis product and MEGA's EA tools provide second-generation collaboration;
3. **Third generation:** third-generation collaboration functionality allows a direct connection to database storage from a Web server. This avoids update delays by publishing through a portal. These tools also enable architects to modify and annotate graphical representations through a Web user interface. The May 2004 release of Proforma's ProServer product allows IT staff to share EA knowledge with all relevant employees across the enterprise.

5.3 Different approaches of EA tools

EA tools are developed for different situations and users. In this section the three main approaches of the EA tools will be discussed. The selection of an EA tool mainly depends on the approach towards the EA. The three kinds of approaches for EA tools are [Peyret2004]:

- **EA top-down approach:** this approach is started at the top (strategies and drivers) and go down to the bottom (technical implementation);
- **EA bottom-up approach:** this approach starts at the bottom and goes to the top;
- **EA change management approach:** this approach starts in the middle to support some projects.

These three approaches leads to three categories of EA tools. Because there is a considerable amount of overlap of functionality, the division between the tools is not strict. The three categories of EA tools are [Forrester2004], [Peyret2004]:

	Top-down	Bottom-up	Change management
Approach	Strategy and planning Applications architecture Infrastructure	Strategy and planning Applications architecture Infrastructure	Strategy and planning Applications architecture Infrastructure
Management	To-be models	As-is models	Projects
User	Business analyst Architect	Operational IT staff	IT portfolio manager IT strategist IT budget controller Architect
Advantage	Provides business process modeling	Assesses change impact and models dependencies	Helps IT prioritize, plan, and budget for business initiatives
Weakness	Hard to maintain accurate descriptions for deployed elements	They're designed for IT and don't link with business processes	Paucity of modeling features

Figure 5-1: The three approaches to Enterprise Architecture [Peyret2004]

- **Top-down modelling:** this category represents the top-down approach. In this approach business process modelling is a necessity and these tools provide then robust modelling capabilities. Some tools in this category started as business analysts' tools for business process modelling, where other modelling capabilities were added later. Other tools were especially designed for EA. Business analysts and architects mostly use these kinds of tools for BPM, data modelling, application modelling, and sometimes systems and networks models. These tools are focussed on describing the To-be architecture (see section 3.2) and the weakness of these tools is that maintaining accurate information for the As-is architecture generates a high workload;
- **Bottom-up repository:** this category represents the bottom-up approach. IT assets are added to a centralized store (the repository) and providing the tools necessary manage updates, to model dependences and predict the impact of changes. Operational IT staff mainly use these kinds of tools because the tools help them to add an application dimension to systems, databases, and networks. This category focuses on the As-is architecture and the weakness of these kinds of tools is that they are designed for the IT. They do not link with business processes, which would be useful in the decision-making process;
- **Change management:** this category represents the change management approach. IT portfolio managers, IT strategists, IT budget controllers and architects mainly use these kinds of tools to prioritize, assess risks and impacts, and track costs at the strategic and operational levels with the help of links between business processes and physical systems. This category focuses on better management of the stages and projects, starting from the As-is model going towards the To-be model. The time dimension of business and IT changes plays a more important role in this category than in the other two. The weakness of these kinds of tools is that they do not intend to supply modelling functionality, as in the bottom-up category. Here the tools will rely on other products for that functionality or provide relatively weak functionality.

With a bottom-up approach a repository can be sufficient to share the EA global view. The As-is architecture can be stored in the repository. The As-is architecture consists of the knowledge modelled by Business Process modelling or development modelling tools. With a top-down approach for business, information, application and infrastructure, and development and deployment with up-to-date models, a complete enterprise architecture modelling suite is needed. For example, an enterprise architecture modelling suite is able to generate UML and data models for the applications by using a common information storage [Schekkerman2003].

In Figure 5-2 a range of EA tools is categorized into the three categories. Some EA modelling suits like Computas Metis, which supports a top-down approach, are able to store different models not only related to EA, but they also provide dedicated customized models to cover EA. Troux, which supports a bottom-up approach, is very strong in its ability to automatically build and maintain the baseline architecture. Alfabet, which supports a change-management approach, is an example of a methodological approach grouping architecture, program, and value management [Forrester2004], [Schekkerman2003].

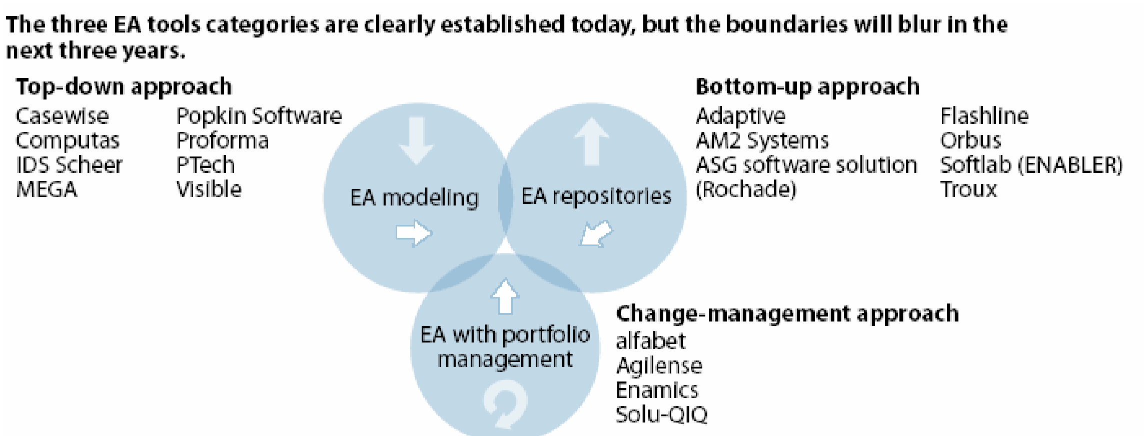


Figure 5-2: The three categories of EA tools [Peyret2004]

Figure 5-2 states that the three EA tools categories are clearly established today, but that the boundaries will blur in the next three years. The top-down modelling tools like Casewise, Computas, and IDS Scheer have included repositories for several years. For instance, the repository capabilities of Metis are very strong. The tools with a bottom-up approach like Troux and Flashline have recently included some change management functions. So the tools are moving closer to each other, which is indicated by the arrows in the middle of the figure [Peyret2004].

6 The situation of Philips

Today, Royal Philips Electronics of the Netherlands is one of the world's biggest electronics companies and Europe's largest, with sales of EUR 29,6 billion in 2003. It all started with the production of incandescent lamps in a little factory in 1891. Under the leadership of ir. Gerard Leonard Frederik Philips, the founder of Philips, Philips developed in the enterprise it is today. In 1922 Gerard Philips resigned as the manager of Philips. During the 31 years of his leadership Philips was developed to an important enterprise with 5,500 employees.

Nowadays Information Technology is seen as a necessary tool for supporting the enterprise and its business process. With the increasing size and complexity of the implementations of information systems, it is becoming more and more difficult to align the information systems to the business of the enterprise. The key problem in the alignment is understanding the enterprise to know which information systems are needed. The complexity and the governance structure of Philips increase each other, which results in a difficult situation to make the right decisions. An illustration of the high complex in information systems in Philips is the current use of about 100 different ERP systems. The management of today cannot understand the structure, processes, and functions of the enterprise as in the time of Gerard Philips. To be able to make sound and responsible decisions, they need an aid that helps them understand the enterprise.

During the past decades the different PD's were independent in their production processes and governance. In fact it begin to seem that there were different independent enterprises that had the same brand name. PD's were doing the same processes by using different information systems, applications or software. Now the management want a closer cooperation between the different PD's. In the field of IT this is done by carefully shifting the scope of Business – IT alignment from the level of PD's to the enterprise level, thus identifying opportunities for synergy and cooperation. For instance all the PD's need an e-mail program. But to be able to identify these synergy opportunities some aid is necessary.

Philips needs an EA tool that helps them to represent their various architectural descriptions on a comparable basis and support the definition of mutual relationships between enterprise level and divisional level architectures. Because of the introduction of the IAF framework, this framework has to be supported in a structural way. The IAF framework was discussed in chapter 4. An EA tool is the solution for supporting the IAF framework in a structured way. The main objective of the EA within Philips is Business - IT alignment. Communication with all the stakeholders (on corporate level as well as on divisional level) is also an important benefit of the tool. In section 3.5 the benefits of an EA tool were discussed.

In the meantime people in Corporate IT are becoming more and more conscious of the coherence between strategy, architecture and governance. Therefore another requirement is that the tool provide in 'traceability of arguments': to make it clear and understandable why a certain decision is taken. With the help of EA Philips has to be able to make better decisions in the field of IT. With EA there is clearness about the reality and this results in a better IT governance. The governance aspect of EA was described in section 3.4.

Based on the theory of section 3.3 it looks like that Corporate IT aims to use EA from an IT-centric approach. Business - IT alignment is the most important objective and decisions based on EA are only taken in the field of IT. The approach within Philips is top-down, because Philips wants to use the EA to go from the top (strategies and drivers) towards the bottom (technical implementation). Also it wants to move the As-is situation towards the To-be situation with the help of an EA tool. This process aspect of EA was discussed in section 3.2 Thus Philips needs, based on the theory of section 5.3, an EA tool that supports a top-down approach.

Enterprise Architecture is a good solution to tackle current problems, such as multiple implementations of the same software solutions, lack of interoperability (between Philips departments and between Philips and its external partners), parallel development of solutions for comparable problems within Philips without parties being aware of that, lack of coordination, etc. Also to support the strategy 'towards one Philips' architecture is needed. The divisions are independent right now and to be able to cooperate in the field of architecture, EA is a good way to support that. But to make EA and the IAF framework of practical use a structural approach is needed. EA tools support this structural approach. If the organization wants to adopt EA enterprise-wide, tools are necessary to make it practical in use.

The next chapter describes the tool selection process of an EA tool that suits Philips to support them in the way that is discussed in this chapter.

7 The selection of an EA tool at Philips

The fourth step to establish an EA, which was described in the introduction, is the selection of an EA tool. That is also the main topic of this thesis and the title of this document is therefore “Enterprise Architecture, the selection process of an Enterprise Architecture Toolset to support understanding and governing the enterprise”. Based on experience, a generally accepted approach to the selection of complex software systems has developed. The ‘best practice’ approach contains a number of stages and activities, of which the most important are [De Heij]:

- The preliminary selection: the preparation of a long list containing less than ten candidates, based on a very limited effort;
- The detailed selection: the limitations of the long list to a short list of not more than three EA tools. This selection is carried out based on a limited number of essential assessment criteria (knock out criteria);
- The assessment: the actual assessment of the short list candidates with the aid of a company specific case during a demonstration;
- The completion: usually focused on entering into contract and preparing the implementation.

The EA tool selection process at Philips followed the approach described above. At this time, however, completion is not realized, due to changing priorities in the corporate IT department. Completion was not part of this graduation period. Section 7.2 reports the preliminary selection in which a long list of eight candidates is composed. In section 7.3 this long list will be reduced to a short list of three candidates. The assessment of the short list candidates, which is described in section 7.4, was done by a demonstration of the suppliers. This phase was followed by a test case with the tool that seems the best after the demonstrations.

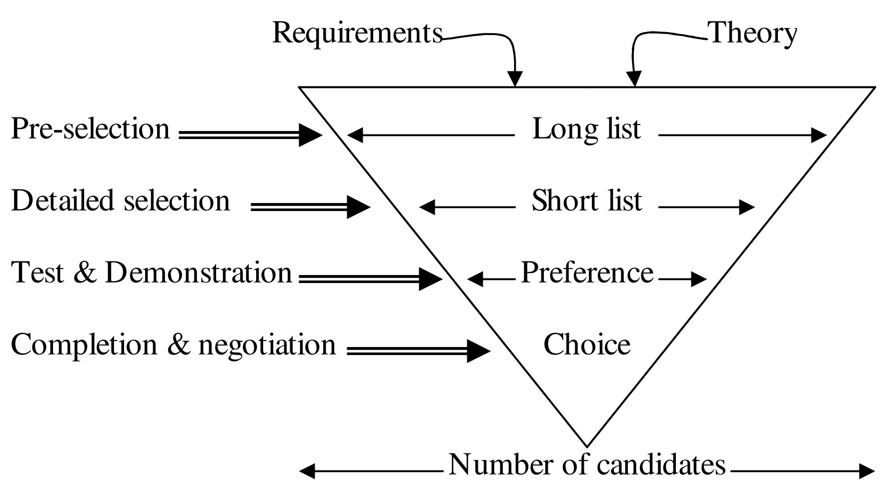


Figure 7-1: The selection process [De Heij]

An important selection aspect, though not the only one, is assessing and comparing the functional features of the candidate systems. In the approach described at the previous page, this is carried out based on user requirements that are then checked using a questionnaire and in demonstrations of the company case [De Heij].

The requirements will be first discussed in the next section. These requirements were used in the Request For Information (RFI). This questionnaire was used to reduce the long list to the short list. Based on the answers to this questionnaire and the demonstrations other requirements were discovered. These requirements will be discussed during the test case. Figure 7-1 shows the different selection stages and the selection was based on the different requirements and the conclusions from the theory from the second part of this thesis. In section 7.1.1 the main conclusions from the theory will be given. These main conclusions were used during the selection process.

One of the first steps in the selection process is to realize that many situations, objectives, and starting points can be distinguished, if only as a result of the size and type of the enterprise. In this case Philips is a large enterprise with a complex multi-national organizational structure. The selected tool has to be able to deal with a lot of users that are all over the world [De Heij].

In essence the selection process is the step-by-step reduction of the possibilities for choice. In other words, at the beginning of the selection process one can choose from the entire spectrum of EA tools that are available on the market, and in the end this choice should have been narrowed down to exactly one option. Considering the breadth of the supply of potential candidates, it is obvious that this process of choosing is taken in a number of steps. A less efficient is to choose by means of a very long questionnaire or an other detailed form of evaluation, where one single candidate is selected from all possible candidates in a single step. For each step a minimum of activities should be performed to realize the necessary reduction, so detailed evaluation should only take place for a limited number of candidate EA tools [De Heij].

7.1 The requirements

In the introduction (see chapter 1) there was a short description of the assignment together with a short version of the Philips' requirements for an EA tool or tool suite. In this section the complete version of all the requirements will be given.

7.1.1 The requirements from the RFI

First the requirements of Philips will be described; these requirements were presented to the vendors in the RFI. The requirements are split up in a section with general requirements and a section with the user requirements. Besides these two kinds of requirements, other requirements were given to the vendors as questions. These requirements were based on the general and user requirements, and the theory from the second part of this thesis.

General Requirements
Philips needs an EA tool or tool suite to model, store, manage and share information about EA. The software tool has to provide a graphical interface to model:

- Business strategy;
- Business activity (value-add business components, processes, organizational units, structures);
- Related information structures;
- Applications to support the business activities and processes (functions, data, systems, interfaces);
- Supporting infrastructures;
- The arguments of how the four models above relate.

The obtained models and information of these architectural areas have to be stored into a repository. An important objective to use an EA tool is 'traceability of arguments': to make it clear and understandable why a certain decision is taken. As a consequence it should be possible to store the arguments in the repository of how the different requirements, principles, and models across the various architectural areas relate to each other.

Philips has decided to use the IAF framework of Cap Gemini as a standard for architectural activities within Philips. It can be expected that changes and or additions need to be added over time; Philips may also decide on some customization of the IAF framework. This implies that the software tool should be able to support the IAF framework and allow customization and adaptation of the repository meta-model.

User requirements
Within Philips architectural activities follow the governance structure. This implies that architectural activities are both executed on corporate level as well as on divisional level. The Philips EA is a set of common principles and models across corporate and divisional architectures. The use of a common software tool must help to represent these various architectures on a comparable basis, and support the definition of mutual relationships between enterprise level and divisional level architectures.

The software tool will have to support both active as well as passive users. Active users are those who enter and maintain models and other elements. In most cases these active users are IT professionals, with a role of architect. Passive users are those who read information and models out of the software tool. They will vary from business managers, to CIO's, to program and project managers and to solution architects. This will require variety in the way architectures are represented towards various target groups.

Architectures are mostly developed in architectural projects, where a team collaborates on the development of principles, scenario's and models. The software tool should support collaborative development. It should be possible to assign parts of an architecture to owners, who have the responsibility to update the formal representation.

Requirements based on the theory
In the theory part of this thesis some requirements to EA and the tools were given. Here only the theory is given what is not described in the general and user requirements. EA is an aid to understand the enterprise. To be able to understand an enterprise its structure, processes, and functions have to be described (see also section 2.1). In this thesis (see also

section 3.1.1), describing the relationships between the different architectures is considered as the added value of an EA. There are different views in describing an enterprise. It is not sufficient to look from one viewpoint to understand the enterprise. Therefore a tool has to take the different viewpoints into account and it has to be possible to add relations between them.

7.1.2 An addition on the requirements of the RFI

The vendors of the EA tool had a few questions after they received the RFI. These questions with answers as sent to all the vendors are presented in this section. Only the questions about the requirements will be described.

Question	Backward Compatibility (Q2.1.4) can be interpreted in two ways: a) functional and or b) data. If functional is meant, than a) the same problem can be treated with a different solution (resulting in learning effort) or b) the same problem should be treated in the same way. What is meant here?
Answer	Philips wants to be able to read the models created in the former (i.e. prior) release(s) into the new release.

Question	1. Give a good definition of the Philips environment to help us understand team sizes etc.; 2. How many enterprise architects are Philips looking to equip? 3. What is the expectation regarding consistency between these enterprise architects and the software development organization?
Answer	**General:** It is our understanding that the application field for tools on (enterprise) architecture on the market can vary: - Prime focus on business architecture/ business processes; - Prime focus on translation business and IT strategy into a IT portfolio's; - Prime focus on describing relationships (traceable arguments/ used principles) between business requirements, application functions, technical solutions. Depending on the characteristics/ strengths and weaknesses of the tool (tool suite), the usage can vary as well. In our RFI we want to get a better understanding of in which area the offered tool is most suited. Our primary goal with this tool is to be able to document in an active way our Enterprise Architecture and its composing Project Architectures and be able to ensure traceability - and impact of changes. In the Philips case we can distinguish the following sub-areas and involved teams: 1. Document the Philips Enterprise Reference Architecture, being the set of principles, standards, and reference models across the layers of business, information, application and technical infrastructure that are applicable for all underlying architectures. Team size: 2-3 people who maintain, published on the website; 2. Document the underlying enterprise architectures on the level of PD-

	s, Corporate Functions and Corporate IT Domains. Various teams; total +/ - 10 people; 3. Document the architecture in major innovation projects, both on Corporate as well as PD level. Total involved team: 10 - 20 architects; 4. Cross-check consistency across the architectures in the three points above here: Involved team: Philips IT Architecture Board (most likely included in the people above).

7.2 The preliminary selection

The preliminary selection was aimed at mapping the supplier's market of possibly relevant EA tools. The result of this phase is a long list of candidate EA tools and suppliers. In practice it is feasible to restrict the long list to fewer than ten candidates [De Heij].

First of all twelve EA tools were selected. During this selection process, the different kinds of tools were taken into consideration and from every category (top-down, bottom-up and change-management, see chapter 5.3) there was at least one EA tool selected. Besides that there were only tools selected that are able to support large companies like Philips. The twelve selected tools, with their suppliers between brackets, are:

1. Adaptations (Adaptive)
2. Alfabet SITM (Alfabet GmbH)
3. ARIS (IDS Scheer)
4. BWise Cubed (BWise)
5. Corporate Modeler (Casewise)
6. Metis (Computas)
7. ProVision - EnterprisePro (Proforma Corp.)
8. System Architect (Popkin Software)
9. Troux (Troux Technologies Inc.)
10. 3D Visible Enterprise (Unisys)
11. Yasper (Deloitte)
12. WebSphere (IBM)

These twelve candidates had to be reduced to less than ten candidates and therefore the following tools were removed from the long list:

- **ARIS:** because ARIS is too much focused on processes. It requires a very formal approach and it is already in use within Philips. In those situations it is shown that ARIS is not suitable to fulfil in the way the EA tool, which has to be selected, has to;
- **BWise Cubed:** because this tool is too much focused on the business processes;
- **Troux:** because it is not clear what this tool can contribute;
- **Yasper:** because it is not a commercial tool. It was only on the long list to consider the thoughts behind this tool and those thoughts can help Philips to formulate their own requirements.

The suppliers of the remaining tools (Adaptive, Alfabet GmbH, Casewise, Computas, Proforma Corp., Popkin Software, Unisys and IBM) were considered in the detailed selection stage. In the next section this stage will be described in detail.

7.3 The detailed selection

The purpose of the detailed selection is to reduce the long list to a short list of three suitable EA tools. Besides this short list, the results of this phase also consist of [De Heij]:

- The information concerning the suitability of the EA tool as regards the critical assessment criteria;
- The supplementary information about the EA tool concerning the history, the number of implementations and references, the future plans and expectations, and the possible hardware and system software;
- The supplementary information concerning the supplier's organization;
- An indication of costs for various components and services.

The 'knock-out criteria' or 'important requirements' are the most important element in the detailed selection approach. These criteria have to be met by the potential candidates to be part of the short list. In principle these criteria are used on the candidates from the long list, after which the most suitable ones remain. If the detailed selection is based on 'important requirements', no more than about fifty requirements should be collected. A larger number of requirements quickly lead to a distortion in the process of choosing. By scoring on these 'important requirements' the short list can be composed. This provides a list of EA tools in order of suitability. Therefore the suppliers have to fill out a short written questionnaire. It is also advisable when one first becomes acquainted with the (for example three) candidates that they present themselves and their EA tool to the project team and respond to some additional questions [De Heij].

During the detailed selection of the EA tool for Philips, important requirements were used. There were no 'knock-out criteria'. Forty important requirements were selected, based on the requirements of section 7.1 and the theory of the second part of this thesis. These requirements were sent to the suppliers as questions in the form of a RFI. The RFI will be described in the next section.

7.3.1 The Request For Information

In the assessment of an EA tool a multitude of requirements play a role. The most important group seems to be that of the functional criteria, though this need not always be the case. From the forty questions of the RFI, nineteen questions were considering functional criteria. There are three main categories of criteria and these criteria concerning [De Heij]:

- **The supplier:** criteria like the size of the supplier, its partners, history, and its financial situations;
- **The EA tool:** criteria like the history and planned future of the tool, functionality, technology, performance, security, and documentation;
- **The conditions of delivery:** criteria like investment, implementation costs, and maintenance and support costs.

For a number of these assessment criteria, information can be provided by the supplier. However, many criteria are subjective and will have to be collected during all the dealings with the supplier, such as demonstrations, presentations, and negotiations. The assessment criteria that were stated, based on the functional specifications, have to be used to evaluate the potential EA tool. This evaluation can be carried out in the following ways [De Heij]:

- By asking the suppliers concrete questions who then provide the answers (preferably in writing). These questions are directly based on the criteria set. Naturally, the supplier is biased, but by formulating the questions as concretely as possible and by including the critical points in the contract, a certain value can be placed on these answers;
- Finding out these things by using the EA tool, i.e. by means of demonstrations, cases, and workshops. Whereas the assessment criteria are directly turned into questions in the other way of evaluation, here the criteria have to be built into the business case.

The first described way of evaluation was used at the beginning of the selection process. The other one was used later on in the final stage of the selection process. The first described way of evaluation was done by sending a questionnaire (the RFI) to the suppliers, which contained the questions about the supplier and the EA tool. Criteria about the conditions of delivery were not considered during this selection process. The RFI was made in cooperation with the Philips Purchasing department. It consisted of three documents and they were made in the tradition of the RFI's of Philips. The three documents are:

- The Main Document;
- The Commercial Questionnaire, which contains the criteria concerning the supplier;
- The Technical Functional Questionnaire, which contains the criteria concerning the EA tool.

The Main Document

The main document is an introduction to the RFI and the selection process. It contains a description of the situation of Philips with regard to EA, the scope, schedule, and process of the RFI. This document also contains the general commercial requirements and a description about how the questions of the commercial and technical questionnaire have to be answered. Besides that, this document contains a form that the supplier has to fill in and reply if it wants to participate in the selection process and a non-disclosure agreement. But the most important part is the chapter about the requirements of Philips towards the EA tool. These requirements are divided into the general requirements and the user requirements from section 7.1.1.

The Commercial Questionnaire

This document is written by René Lenarts, who is from the Philips Purchasing department. The questions in this document are about the supplier of the tool or about other current users of the tool. The answers to these questions serve to verify the continuity of the supplier.

The Technical Functional Questionnaire

This is the most important document of the RFI. In this document the technical and functional questions for the suppliers were formulated. Preparing and assessing this questionnaire was quite a task. Suppliers often interpret the questions in different (sometimes creative) ways. In section 7.1.2 there are examples of questions that were unclear for the supplier. Moreover, often the requirements of the questionnaire demand specific solutions in the EA tool while different options, which might not yet have been thought of, are also possible and sometimes better. On the other hand, a supplier confirming that he can offer a solution to the defined problem does not yet guarantee any quality [De Heij].

Example from the RFI: "Is it possible to generate a proprietary framework?"; a "Yes" of the software supplier does not say anything about the way this supported, how it can be generated and the simplicity of use. Therefore the Yes/ No questions were built up in the following four parts:

- **Question:** contained questions like in the example, which can be answered with yes or no;
- **Comments:** here additional questions were asked, e.g. 'explain how it works';
- **Answer:** here a yes or no answer was expected;
- **Explanation:** here additional explanation could be given, e.g. how it works and how easy it is. This gave the vendors the opportunity to give extra information, which could result in describing possibilities of EA tools that were not considered before.

The answers of the suppliers of the EA tool to these questions were the decisive factor in selecting the tools for the short list. The questions were divided into the following four categories:

- **General information:** the questions in this category were about the tool or tool suite that the supplier offered to meet the requirements of Philips as presented in section 7.1.1;
- **Functionality:** the questions in this category were about the functionalities of the tool. The questions were for example about the companies that already use the EA tool, the frameworks the EA tool supports, the modelling techniques the EA tool provides, the import- and export facilities, the industry standards, the aspect views that can be created, whether the EA tool helps in finding synergy opportunities and whether the EA tool allows traceability between the transformations;
- **Technology:** the questions in this category were about the repository, the licences, platforms, whether the tool is web based, etc;
- **User requirements:** the questions in this category were about the user interface, language, and training.

7.3.2 Analyzing the response of the RFI

In all phases an objective consideration should always be attempted. For this purpose the assessment criteria have to be made fully explicit. The assessors have to evaluate the EA tool (and the supplier) using these objective criteria. If the criteria are not explicit, vague valuations arise that say more about the communication and commercial skills of the supplier than about the ability of the EA tool to fulfil the objective requirements. Therefore in this section the valuation criteria, which were used in the selection process, will be described. Also the conclusions towards the feedback of the different suppliers to the RFI will be given [De Heij].

The offer of IBM was not considered because by leafing through the RFI it was clear that the offer of IBM was not what Philips is looking for. It was too much focused on the development of new software intensive systems. Philips is looking for an EA tool that also considers the process, which takes place before the development of a system, like business processes. Kees Wielinga and Frank Butstraen were also of opinion that the offer of IBM is not what Philips is looking for. Therefore it was superfluous to analyze the RFI of IBM.

Unisys decided not to bid on the RFI, because they are not a real EA tool vendor. Therefore it was impossible to analyze Unisys in the same way as the other tools. Instead of answers to the RFI, Unisys did send in all kinds of information of what they offer for EA. During this analysis Unisys was not considered, but Unisys was compared with the top 3 that was based on this analysis of the different feedback to the RFI. After the comparison it was considered that Unisys does not provide that what was expected of the other tools.

To analyze all the feedback to the RFI, a document was made that consists of the following five parts:

- **Introduction:** this part was an introduction on the analysis of the feedback to the RFI. In this part the analyse method of the feedback, the requirements of Philips, and the most important aspects of the analysis were described;
- **Questions:** in the analysis of the feedback to the RFI only the answers to the questions of the Technical Functional Questionnaire (from the RFI) were considered. In this part of the review document all those questions of the RFI were given;
- **Review:** all answers were rated. The companies that were considered in the analysis were: Adaptive, Alfabet GmbH, Casewise, Computas, Proforma Corp. and Popkin Software.
- **Remarks:** per question it is explained what the review focused on and how the answers were judged;
- **Conclusion:** for each tool a concise assessment was formulated. For the selection only the answers were considered that were considered as the most important ones in comparing the different tools. This was done to divide the necessary requirements from the optional ones. Based on the necessary requirements a better decision could be taken. The concluding advice was to select Adaptive, Casewise and Computas for the short list.

Introduction
The requirements were described in section 7.1. This section contains a description what is considered as the most important aspects of an EA tool. On these aspects, there was a focus during this review:

- Philips wants to be able to document in an active way their EA and its composing Project Architectures and be able to ensure traceability and impact of changes;
- Philips needs an EA tool to deal with all kinds of architecture in a structured way. In the beginning this architecture will takes place in small projects but the purpose is to use EA more and more in the future. Therefore the EA tool has to be able to reuse obtained results in the future;
- In this review there was especially paid attention to the possibility of creating different views, the relationships between the different models and views, the usage of the IAF framework in the different tools, the different modelling tools that can be used in combination with the tool (like Visio), the import and export facilities (like the use of XML), and if the tool can cope with different users in different places (if it is web-based and has a central repository).

Review
The answers of the six remaining tool vendors were rated on a scale of one to three. A 'one' was given if the answer suits the situation of Philips and a 'three' if it does not fit the

situation of Philips. Of course questions about the name or version of the tool were not rated this way.

The top three, with between brackets its obtained points, of each category was:
- **General information:** Computas / Proforma / Adaptive (3);
- **Functionality:** Computas (25), Casewise (26) and Popkin (27);
- **Technology:** Computas / Proforma (11), Adaptive / Casewise (12);
- **User requirements:** Popkin (5), Adaptive / Casewise / Computas (6).

An extra mark was added reflecting the general opinion about the tool. That mark is based on the total RFI together with the additional material the vendor sent. Four vendors got a 'one' (Adaptive, Casewise, Computas and Popkin), one vendor a 'two' (Proforma) and Alfabet got a 'three'. The total result was: Computas (46), Casewise / Popkin / Adaptive (50). Alfabet (74) and Proforma (58) were clearly not the tools that are suitable for Philips.

Further selection
For a good analysis of the feedback to the RFI some questions were selected. It was considered that these questions addressing the most important requirements of Philips. To consider only the most important requirements the tools will be judged on the absolute important aspects for Philips. That gives a better judgement than considering all the questions in which additional functionality (which is considered as nice but not necessary) can mask some drawbacks of the tool.

The key questions considered aspects like modularity of the tool, what the tool does, who the intended users are, the IAF framework, the import- and export facilities, the relationships, the different views, synergy options, and if the tool is web based. Also the extra mark was taken into consideration.

This evaluation resulted in the top three: Computas (21), Casewise (23) and Adaptive (25). To conclude this section a judgement of the offer of each vendor will be presented as well as a general conclusion.

Conclusion per vendor:

Adaptive
Adaptive has relationships between the different models and it looks like handy in use. It uses tools like MS Visio and MS Office. Other strong points are: it is web-based and it uses XML. But after the RFI selection process the capabilities and possibilities of the tool were not fully clear. However the tool could be suitable for the situation of Philips.

Alfabet
Alfabet is too much focused on portfolio management of IT projects. Its purpose is to group EA models with IT management models and it is an example of a methodological approach grouping architecture, program, and value management. Alfabet does not take the different architectures with their mutual relationships into account. Therefore it is not suitable as an EA tool for Philips.

Casewise
Casewise offers a promising tool especially because it has relationships between the different models. It is a comprehensive tool with a lot of options and also a strong point is that it works with standards like XML and BPMN.

Computas
Metis from Computas is a very promising tool. Its strong point is that the user can use different views (thereby he is not limited to predefined views) with relationships between the elements in different model types and diagrams. It also uses standards like XML, BPMN and UML and the models look very well organized because Metis integrates all the models into holistic enterprise-wide model structures.

Popkin
Popkin has a lot of options and modelling techniques, but is not convincing on the subject of relationships. They claim that their tool supports the Business - IT alignment, but the impression arose that they only fill in a framework (like Zachman) without considering the relationships between the models. The feedback to the RFI is, compared to the other ones, not convincing. Although the tools of Popkin are widely used and acknowledged (they often appear in articles about EA) the analysis concluded that there are better ones.

Proforma
This tool looks good and it looks very easy in use. But an important drawback of the tool is that there are no relationships between the different models. Therefore this tool is not considered as sufficient to the needs of Philips.

Summary of analyzing the feedback to the RFI
The tool has to be able to generate and store different models with relationships between them and in harmony with the IAF framework. In the beginning the tool has to support small architectural approaches, but in the future it has to support all the architecture within Philips and also has to make 'what if' analysis possible. Taking this in consideration Metis from Computas and Corporate Modeler from Casewise were considered as the best tools for Philips. In section 5.3 the tools of Casewise, Computas and Popkin were considered as top-down (which is the case in the situation of Philips) and the tool of Adaptive as bottom-up. To try out a bottom-up tool it was suggested, also on the basis of the RFI, to select Adaptive as the third tool for Philips.

7.3.3 The Short List

In the previous section the responses to the RFI were analyzed and the top three were selected for the short list. Thus the selected tools are Metis (Computas), Corporate Modeler (Casewise) and Adaptations (Adaptive). As mentioned in the introduction, these three suppliers were invited to give a presentation and a demonstration of their tool. These presentations and demonstrations will be described in the next section.

7.4 Demonstrations

Based on a specific business case, consisting of realistic data and business processes, all (in this case three) candidates had to give a comparable demonstration. The demonstrations were meant to assess the candidates on the short list on important aspects. But also the demonstration gave a good opportunity to see how the tool works. Also it gave proof if the tool can execute what the vendor claims and how the tool does that. This resulted in a ranking and the top candidate of it will be selected unless inaccuracies are uncovered in the next phase or the result of negotiations is negative [De Heij].

The vendors of the three tools of the short list (Computas, Casewise and Adaptive) were invited to give a presentation and a demonstration of their tool. The selected business case was a part of the PHilips Enterprise Reference Architecture (PHERA) model, with business strategies, business processes, IT strategies, and applications. The vendor's presentations and demonstrations will be summarized in the following sections. After the presentations and demonstration the tool that looked like the best one for Philips would be tested.

7.4.1 Computas

Computas gave a presentation and demonstration of their tool Metis. It was a good demonstration in which they emphasized the possibilities of creating relationships between models and principles in Metis. Also they made different views of the same situation and showed how to make comparisons between different views. In Metis containers are used (for example a process) and it is possible to zoom in on a container and find the processes behind the process. With the help of queries it is possible to get all kinds of information like which application is needed for which process. The graphics in this tool are very impressive. With the graphics it does not look very difficult to work with this tool. The presentation and demonstration supported the conclusion of the RFI-analysis that Metis is a promising tool, which can meet the most important needs of Philips.

7.4.2 Casewise

Casewise gave a presentation and demonstration of their tool Corporate Modeler 9. During the presentation it became clear that Corporate Modeler, despite its use of relationships and different views, is not as good as Metis. The comparisons between different (part of) models look better in Metis than in Corporate Modeler. After the demonstration with Corporate Modeler 9 they showed Corporate Modeler 10. The new version is graphical much and much better then its predecessor. This version is definitely an improvement on the graphical interface. The functionality is not changed, so the drawbacks of Corporate Modeler 9 also apply on Corporate Modeler 10. With this new version it looks not very difficult to work with the tool.

7.4.3 Adaptive

Adaptive is a repository-based tool while both Metis and Corporate Modeler are modelling tools. The repository of Adaptive is the most important part of their tool. It is an open and completely standards-based (MOF Compliant Repository). A drawback of this tool is that in order to use this tool every time a connection with the repository has to be made. An advantage of this tool is that it has a standards-based integration with other tools like Metis and Visio. Also it has a very good existing meta-model in which strategy is linked to objectives. The graphics in this tool are somewhat disappointing. It does not look very difficult to work with this tool, but the pictures and layout are distressingly bad.

7.4.4 Conclusion

After the presentations and demonstrations a decision was made. Metis and Corporate Modeler have the same functionality, but Metis is considered better for the situation of Philips. It has a better user interface and more additional options. It does look easy in use and the relationships and the comparisons between different (part of) models are better than in Corporate Modeler.

The biggest advantage of Adaptive is its repository. In the field of the graphical interface Metis is definitely in favour. Also the conveniently arranged models in Metis are an advantage with regard to Adaptive. Relationships in Metis are clearer than in Adaptive and all the relationships, that are possible in Adaptive, are possible in Metis. Other functionalities of Adaptive or equal or less compared with the functionalities of Metis. Because Metis also uses a repository, Adaptive was not longer considered as the best tool for Philips.

If it appears from the test case that Metis can handle version control and authorization than Metis is the best tool for Philips. These requirements were not considered before and therefore they were not part of the demonstration of Computas and Casewise.

After the three demonstrations, the judgement of the EA tools, which was based on analysis of the review to the RFI, was not changed. Therefore Metis was selected for the test phase.

7.5 The test case

The test case was only held with Metis. As mentioned before, there were no comparisons possible with other tools during this test case. The purpose of the test case was to find out how Metis is able to support the requirements for the test case are described in section 7.5.2. These requirements were based on the requirements from section 7.1, the requirements based on the various feedback to the RFI, and the requirements based on the three demonstrations. The requirements based on the feedback are requirements that were not considered before the RFI, but what are also requirements for the situation of Philips.

During the test case the architecture of the PD Semiconductors (SC) was implemented. That was done by implementing three different views and with the help of these three views a fourth view was made. The fourth view is based on the IAF framework.

The main purpose of the test case was to find out if the tool is able to check consistency, links and overlap between different views of the same architecture. During this test the consistency, links, and overlap of the architectural views between the Hydra and the PHERA document were checked. Therefore the As-is reference architecture of Hydra and PHERA were implemented in the EA tool. The owner of the Hydra view is the product division Semiconductors (SC) and the owner of the PHERA view is the PHERA team, which consists of architects from every PD and corporate IT. Also a third view was created. This is the SITe ARchitecture (SITAR) view and from the SITAR document things like strategies, business principles and IT principles were taken. The owner of the SITAR view is the SITAR team. The aim of SITAR was to define a reference architecture, which could be used at all the Philips sites. After implementing these three views a fourth view was created. This view is based on the IAF framework and the input was generated from the PHERA, Hydra and SITAR views. These four views were the input for the different test scenarios of section 7.5.2. To execute some of these tests it was necessary to add elements, which are not based on the real situation within Philips. For instance there were created some links between strategies and business processes. Obviously these links exist, but they were not documented. These links were created to do some analysis such as special search criteria. The results of the test case were presented during a meeting of the Architecture Committee of Philips. During the presentation a demo of Metis (with the used business case) was given.

7.5.1 Introduction to the business case

In this section the business case, which was used during the test case, will be explained. Figure 7-2 shows a graphical representation of the architecture of Philips. The box on the top with EA in it represents the corporate architecture of Philips. The SITAR view belongs to the corporate architecture of Philips. Under this corporate level all the PD's are visible. The first box is filled in and this box represents the architecture of the PD SC. In the case of SC this architecture is called Hydra. Also the corporate functions, like corporate IT, have their own internal architecture. The PHERA architecture describes, on corporate level, all the architectures of every PD in a common style. The Hydra architecture and the PHERA architecture are drawn in Figure 7-2. In this section both views will be described shortly. The Hydra architecture is only used in the PD SC and therefore only the SC part of the PHERA architecture was implemented in Metis. In Figure 7-2 a question mark is shown. An objective of the PHERA document was to identify which parts of the divisional architecture could become a part of the corporate architecture in the form of shared services. The question mark stands for identifying these shared services.

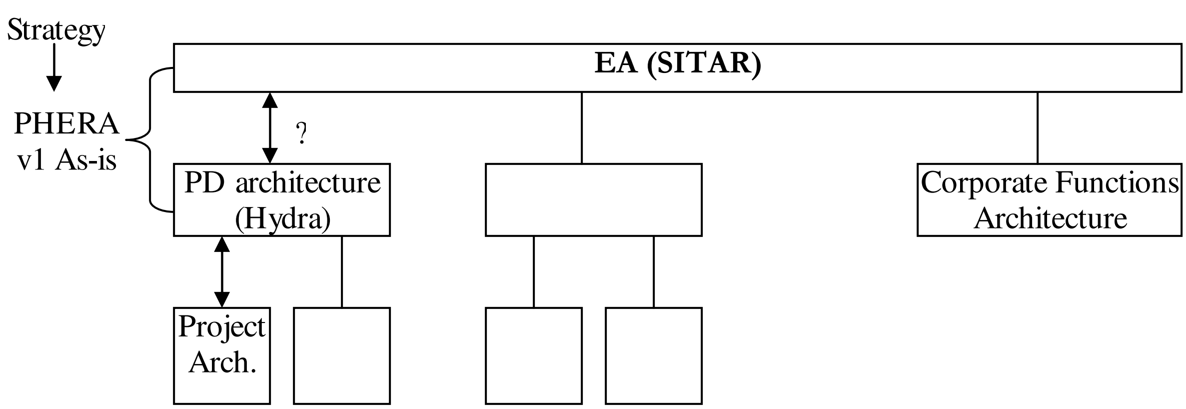


Figure 7-2: The architecture of Philips

PHERA

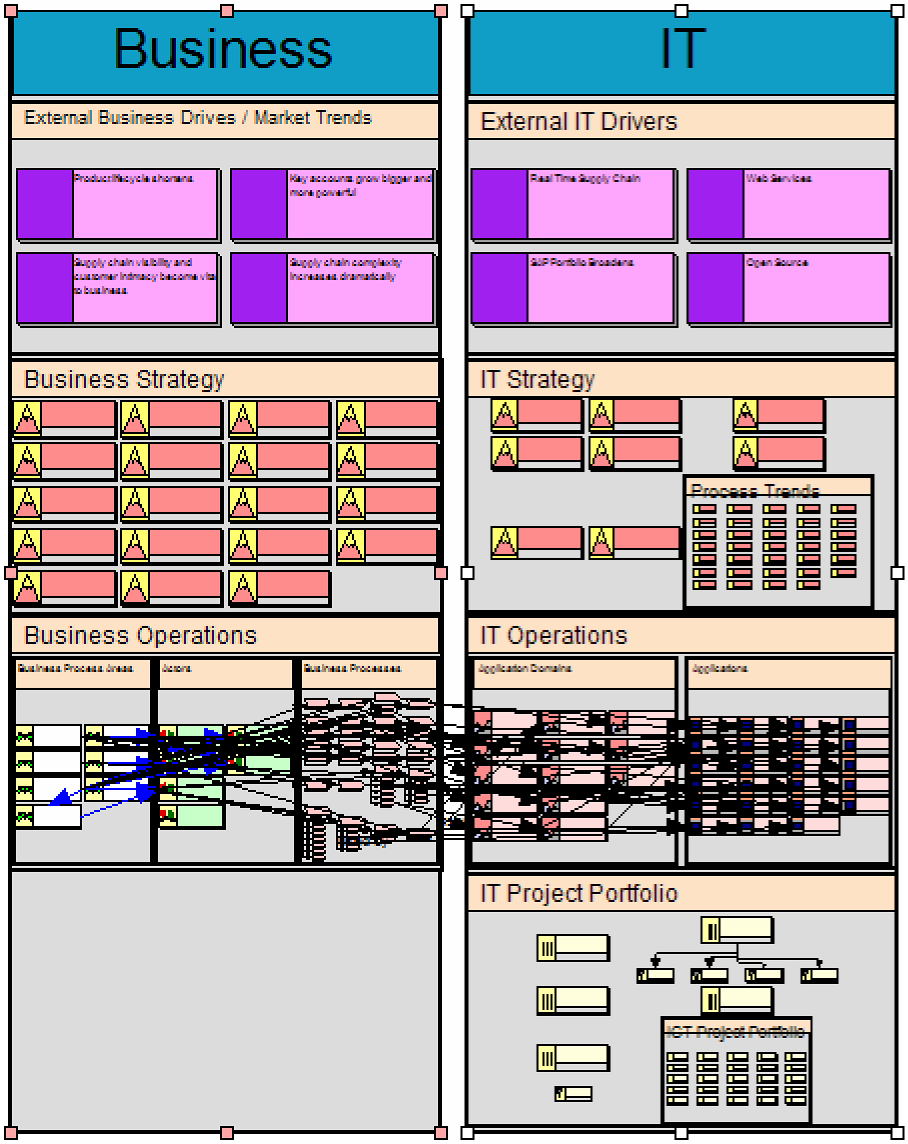


Figure 7-3: The PHERA view imported into Metis

Figure 7-4 shows the PD Analysis model that is used in the PHERA document. This view was implemented in Metis and compared with the view that is represented in the Hydra document. The principles and especially the relationship between the business and IT principles were compared with the situation in Hydra and SITAR. The PHERA document contains tables with information such as business drivers, business strategy, business processes, IT drivers, IT strategy, applications, and IT project portfolio. This information was imported into Metis and represented as objects with relationships between them. In Figure 7-3 a screenshot of the PHERA view in Metis is given.

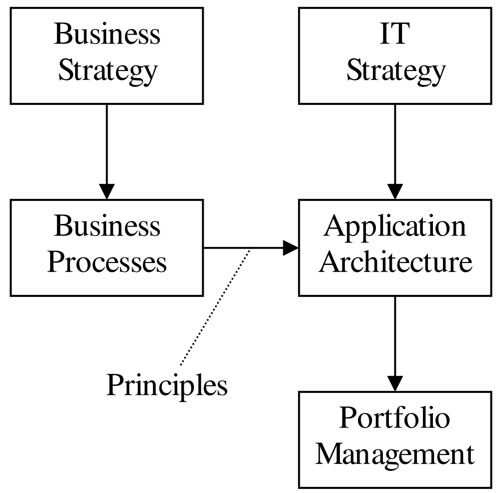


Figure 7-4: The PD Analysis model of PHERA

Hydra

Figure 7-5 shows a part of the PD Analysis model that is used in the document of the PHERA team. This part of the PD Analysis model had to be filled with the data from the documentation of Hydra. The arguments of the relationships between the business processes and the application architecture had to be compared with the relationships between the principles of the situation in PHERA and SITAR.

Figure 7-6 shows the Hydra view, which was used in the PD SC. The figure shows the domains together with their applications and the relationships between the different applications and domains. The figure shows how the different domains are presented and this overview was also implemented in Metis. Besides the overview from Figure 7-6, there was a document with additional information about the Hydra view. The arguments were taken from this additional information.

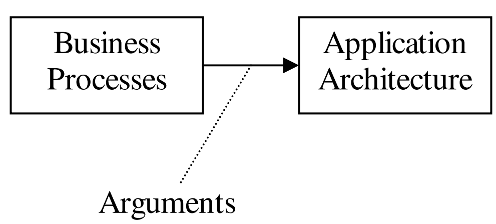


Figure 7-5: The Analysis model of PHERA, which had to be filled with the Hydra information

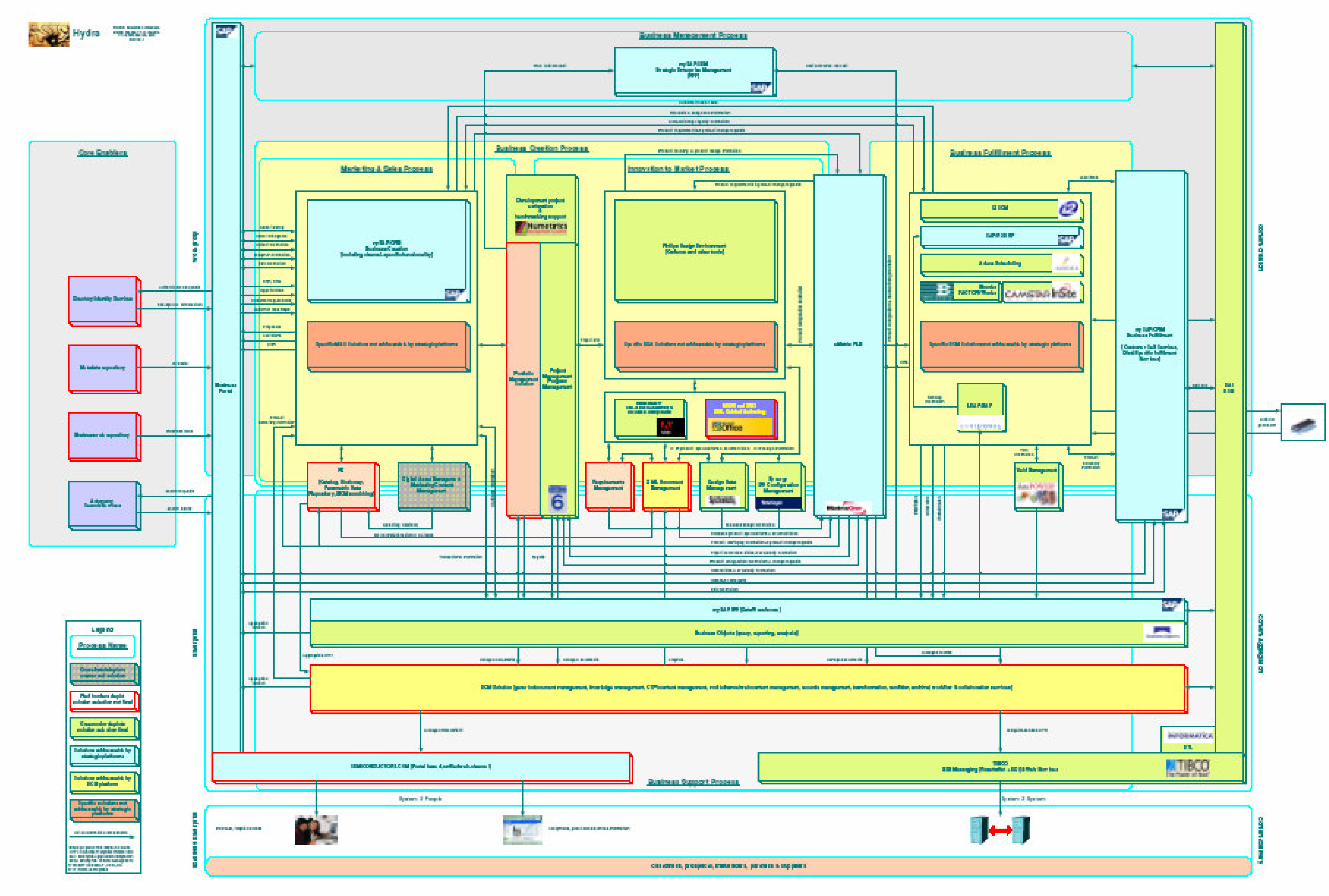


Figure 7-6: The Hydra viewer

SITAR

From the SITAR document the business principles and IT principles were taken. These principles were divided into different areas in the document and they were also connected with the right areas in Metis. These principles were compared with the principles of the PHERA document. The SITAR document contains a lot of tables in which all the IT principles and business principles are introduced. In another table these principles are linked to each other. It is very hard to find all the IT principles that belong to a certain business principle. After importing all these principles, with their mutual relationships, into Metis an easy overview was generated. Figure 7-7 is a screenshot of the SITAR view in Metis. The relationships are represented as arrows to see which object influences another one. In Metis it is possible to hide all the relationships or to select make some relationships visible.

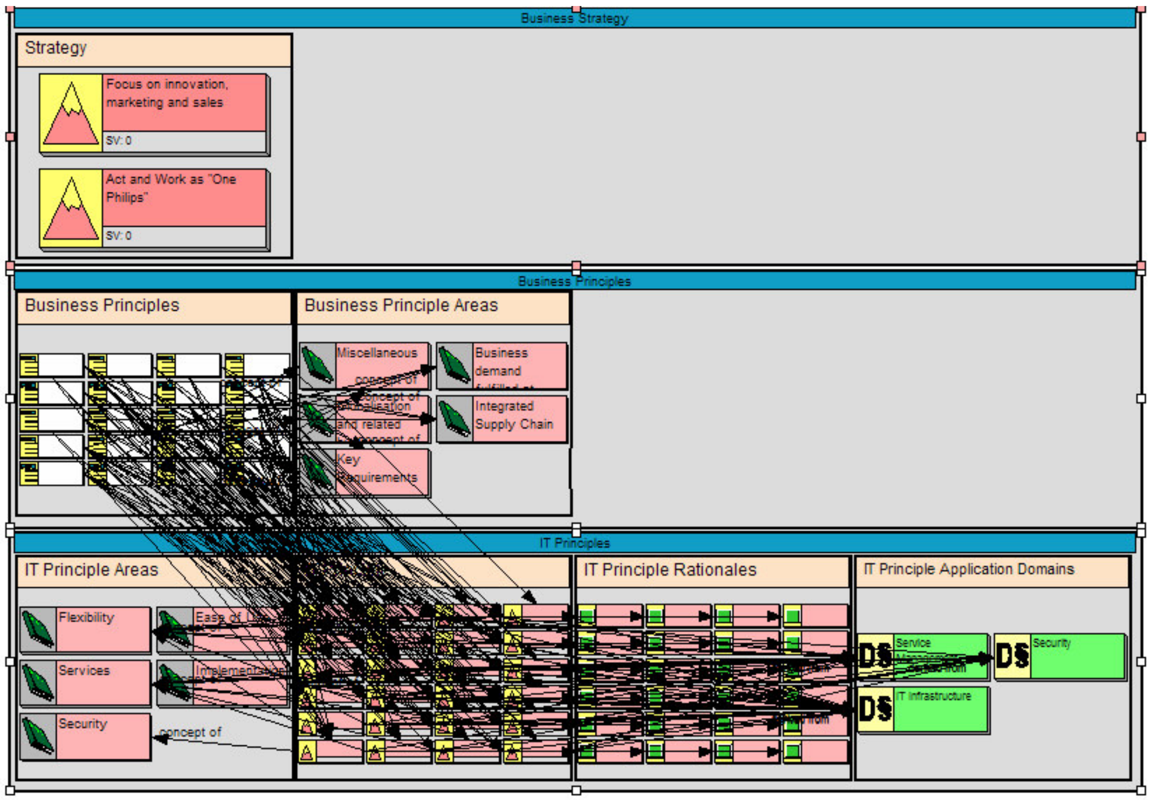


Figure 7-7: The SITAR view imported into Metis

IAF
The input, from which the IAF view is composed, is from PHERA, SITAR, and Hydra view. The input from these views will be placed into a cell from the IAF framework. Thereby the cell will be divided into three areas: one for PHERA, one for Hydra, and one for SITAR. This results in a clear overview from what view the input is taken and this makes comparisons between the different views easy. After generating the input, the relationships between the objects (from the different views) that represent the same, were created. Figure 4-7 shows a screenshot from the IAF view, which was imported in Metis.

7.5.2 Test scenarios

This section contains a short overview of all the different test scenarios that were implemented or dealt with during the test case. The sections are in order of importance.

Implementing the different views
As mentioned before the purpose of the test case was to find out if the tool is able to check consistency, links, and overlap between different views of the same architecture. During this test the consistency, links, and overlap of the architectural views between the Hydra and the PHERA document were checked. Also the arguments of the relationships, between the business processes and the application architecture, in the Hydra view had to be compared with the relationships between the principles in the PHERA view. Other aspects that were tested in this phase:

- **Processes:** how is it possible to create processes;

- **Relationships:** how are relationships built and how many different relationships can be implemented;
- **Principles:** how are principles and the relationships between them implemented in the tool;
- **Properties:** which properties of the principles, processes, and objects can be imported into the tool;
- **Import pictures:** is it possible to import existing pictures, how does it work and is it possible to edit the pictures e.g. with links;
- **IAF:** how is it possible to implement a fourth view based on three existing ones?
- **Models:** is it possible to import complete models, how does it work and what are the requirements for that;
- **Strategy:** how can the strategy be implemented and related to the principles and models;
- **Arguments:** why certain decisions were taken in the past leading to the current As-is architecture.

Authorization
Is it possible that different users have different kinds of access per model? Are there the following three kinds of users or access levels:

- Owner: is responsible and is allowed to make changes;
- Viewer: is allowed to view the architecture but is not allowed to change it;
- No access: is not allowed to see or change it.

Does a simulation consider also parts of the architecture that the user is not allowed to see? Is it possible that an owner is only responsible for a small part of the enterprise and that he is allowed to see some other parts of the enterprise? When an owner makes a change (in model or principle) how does that reflect on the views that are built by others? Is it possible to see the impact of changes?

Version control
How are the differences between the As-is architecture and the To-be architecture visible? Is it possible to link the portfolio management to the gap between the As-is and the To-be architecture and how does it work?

Is there a traceability of arguments possible and how does it work? How is it possible to make it clear and understandable why a certain decision is taken?

Search options
On which ways is it possible to find elements, relationships or processes? Is it possible to find it by giving the name (or a part of the name) of what you are looking for? Or is it possible to find it by browsing through the models?

Is it possible to find on which elements the process or principle has influence or which elements influence the process or principle? How does it work?

Simulation
What can be simulated and how does the simulation works?

Is possible to create an impact analysis?
Is it possible to create a cost analyse? How does it work and what does the meta-model look like.
Are there also SQL queries and how does the database schema looks like?

Synergy and Redundancy
Is it possible to find synergy opportunities and how?
Is it possible to find redundancy and how does that work?
On which ways is it possible to compare different business processes?

7.5.3 Conclusions of the test case

This rest of this section is divided in four parts. The conclusion, based on the test case, were about:

- **Metis:** the conclusions about the tool itself;
- **The test criteria:** the conclusions about the answers to the test scenarios;
- **Conclusions of analyzing Hydra and PHERA:** what are the results of implementing these architectures and which gaps are there or which information is still missing?
- **The contribution of Metis for Philips:** the conclusions about what this tool can contribute in the situation of Philips.

In this section a couple screenshots will be given. These screenshots are from the implementation in Metis, which was created during the test case.

Metis: general conclusions
Metis is a nice and easy EA tool. In only a few days it was possible to create a good-looking AD. It works very nicely because it uses a lot of standard Microsoft commands such like Crtl c and Crtl d.

The tool contains a standard meta-model, which is based on a lot of experience in other enterprises and is developed in close connection with these enterprises. But if it does not fit the enterprise, it can be changed. To do that a certain license is obligatory which was not available for us, so this could not be tested during this test case.

Metis can also generate extensive output, rapports or searches. In Figure 7-9 a budget report is drawn. This report was created with Metis and it was based on the information implemented in Metis. The more complicated the output the more difficult it will be to create it. But in Metis experienced users will be able to generate all kinds of different output, because it works can be generated in a straightforward way.

In Metis data can be imported from excel files or directly from databases. So it is possible to reuse data that already exists in the enterprise.

In this test I discovered not much shortcomings. The help file was not very useful when I needed it. There are a lot of options in the tool and sometimes you are lost in all these options.

It is possible for users to create their own AD in the way they want that it looks like. Graphical a lot is possible, for instance all kinds of pictures can be imported and buttons can created. By creating the same graphical overview, as Philips' employees are used to, they get comfortable with Metis very soon. By clicking on the buttons it is possible to surf to another view in Metis, or to zoom in into an area.

In conclusion Metis is a very nice tool to work with and in during this test case all functional and user requirements are met.

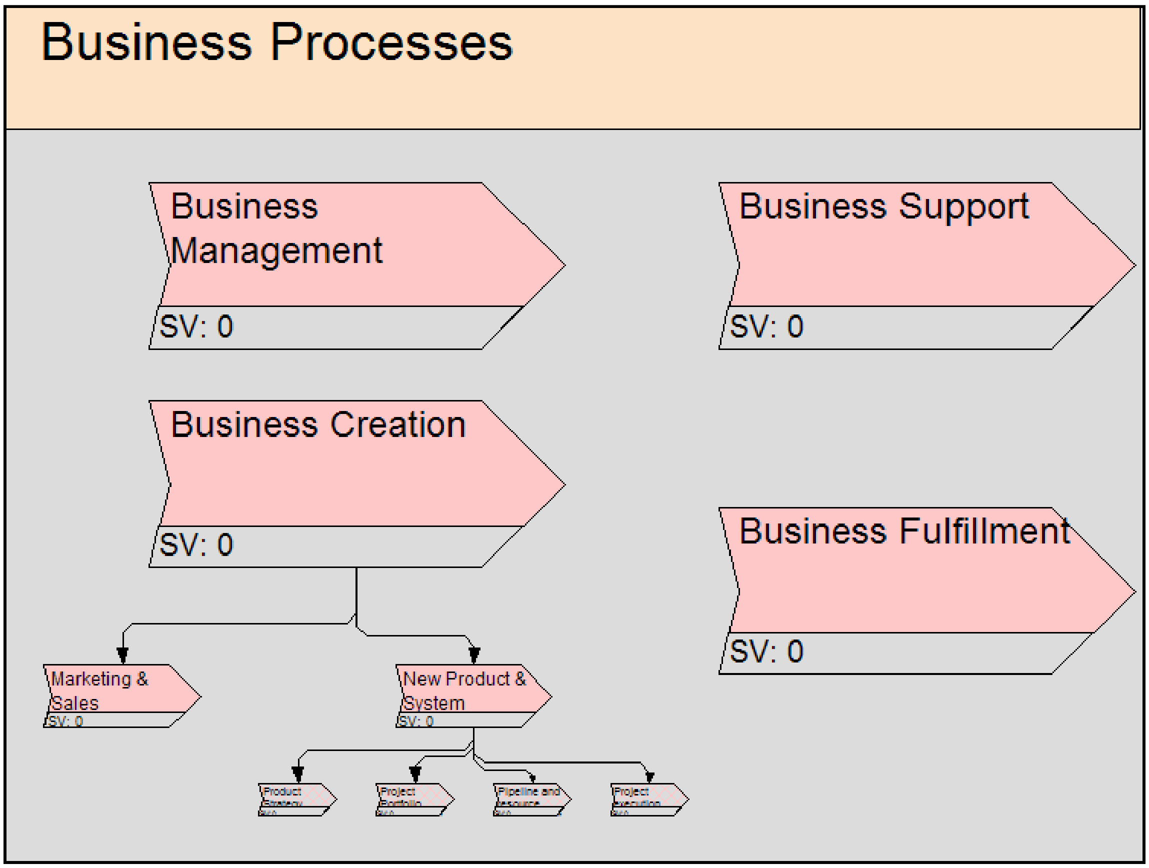


Figure 7-8: A screenshot of a container with Business Processes in Metis

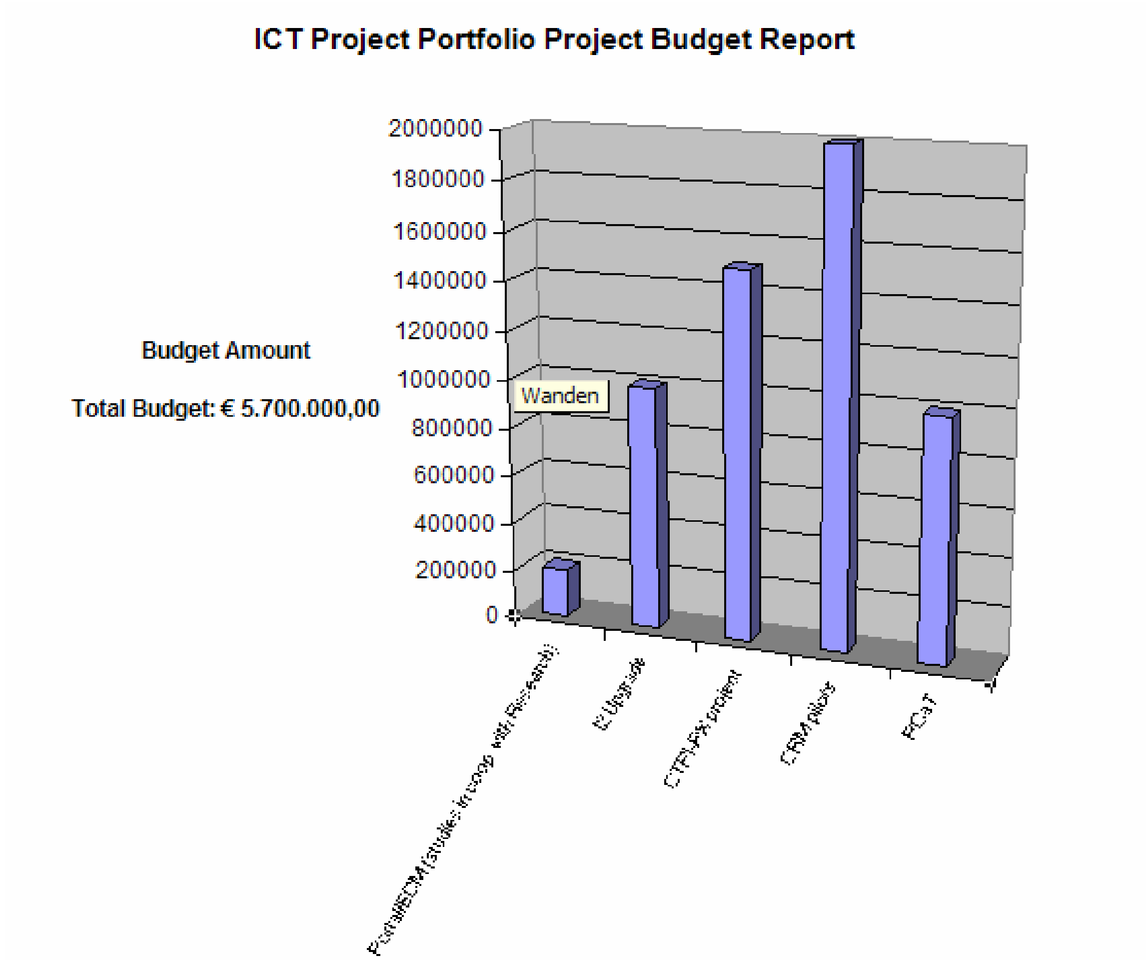


Figure 7-9: A budget report in excel created by the input in Metis

Test Criteria
It was proved during the test case that Metis can handle the most important aspects of EA like creating different views and relationships between these views. The requirements that caused questions or problems during implementing the test scenarios were:
Warning of change: When there are different views, which are based on the same data, they all will be changed automatically if the original data is changed. A warning does not automatically appear. When it is needed to warn someone this can be done with the annotations. Annotations are messages that can be stored with the different models. A user has to open the annotator view to see if the owner of the model has left a message. This message informs him about the change made by the owner. A more active form of warning would be appreciated;
Changing an object type: It is possible to change the object type of one or more objects. When the object type is changed the relationships and names will be left intact. That is very suitable and saves a lot of time;
Creating the IAF view: The requirement here was to create a view based on the IAF framework, which does not affect the other views. But if there is made a change in the other views they affect the view based on the IAF framework. So if someone makes a change in

the Hydra view, it is also visible in the view based on the IAF framework. But when there is made a change in the IAF view, nothing should happen in the Hydra or PHERA view. This is not exactly possible in Metis. But there is a sort of solution to create this. The objects from the PHERA or Hydra view that you want definitely in your IAF view have to be copied and paste (with the container) as a dependable view. Now if something changes in Hydra it also will be changed in the IAF view. Secondly you create in the IAF view another container, with the same name as the copied container and these containers can be linked to each other. So now there are two containers: one that is dependent on the other views and one that is independent.

Conclusion: Metis can achieve the most important all the test criteria of the test case and therefore it succeeded in the test scenarios.

Figure 7-10: The relationships of a Business Process in Metis

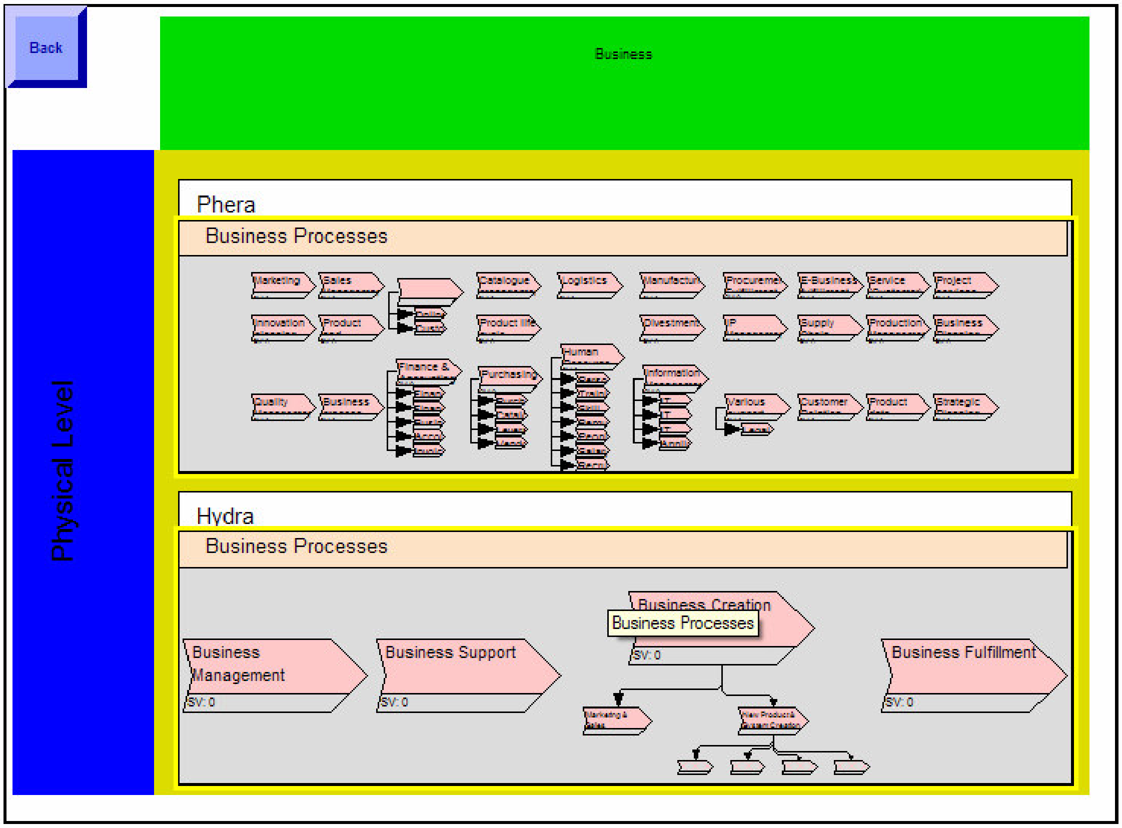


Figure 7-11: A comparison of the Hydra and PHERA view in a cell of the IAF framework

Conclusion of analyzing Hydra and PHERA
During the test case the Hydra, PHERA, SITAR and IAF view were implemented. The conclusions of the analysis are:

Hydra and PHERA are not consistent. They contain different areas and processes for instance. In Figure 7-11 there is a comparison between the processes of PHERA and Hydra. After presenting this as a result it was said that there are company policies for this. In both views these policies were not followed completely. Metis supports architects to find inconsistencies between various architecture models, or -specifically- between their own model and the corporate model that they should be consistent with. This will lead to further consistency between Philips architectures company-wide, which is important for finding synergy opportunities and for integration initiatives.

Hydra is not complete. In compliance with the models that PHERA provides, certain processes in Hydra should be split up in other processes or some processes should contain subprocesses. Metis enables to compare the different architectural views to see if they are consistent. By describing process on the same level of detail, Philips will be able to spot opportunities for shared development, reuse of existing applications or to use the experience of one division to improve implementations in comparable projects.

Hydra definitions are not strict enough. In Hydra 'process bridges' are introduced, but they are not defined in such a way that they can be described in Metis. Therefore the process bridges had to be defined better before they could implemented. Metis is very strict in types and definitions, so by implementing an architectural view everything has to be defined in a good way. In Hydra process bridges were a set of processes and applications and these process bridges were a link between two business areas. In Metis it is now defined as a Business Function. That is a high-level business entity, which can be considered as a high-level business process. Metis is here used as a reference language. For Philips this is very useful, because a reference language is important in case of collaboration between PD's, mergers and acquisitions, strategic alliances, and other forms of business cooperation. With Metis Philips has a reference language without spending much time and, effort creating one.

By using the IAF framework, relevant business questions can be generated. Because the view of the IAF framework is based on all the other views, with relations between the same objects of the different views, the business questions can give more specific answers. The SITAR document contains principles and strategies. By combining these principles with the processes and applications from the PHERA and Hydra view, dependencies between strategies and applications can be identified. With a tool as Metis it is easy to find these dependencies. For instance, during the test case the strategies that were supported by the SAP implementations were easily found by special search criteria. In documents, which are used today, it is only possible to find these by creating a lot of tables that contains all kinds of dependencies. The strength of Metis is that all kinds of business questions can be generated and that the results are directly visible.

Contribution of Metis for Philips
There are many employees that are responsible for different parts in the organization. Metis takes these different responsibilities into account. The most important aspect for Philips is to create an architectural view of their PD's and corporate IT, so they can cooperate better in the field of architecture. After implementing the different views portfolio management is possible. Also a better cooperation in the field of architecture between the different PD's can become a result of using Metis. Metis also can lead to a better Business - IT alignment in Philips. Metis is able to directly support the 'towards one Philips' approach by providing a tool to find synergy opportunities. For instance the PHERA and Hydra view could be compared in an objective way. This also possible between different architectures to find similarities and to create a To-be architecture where the different aspects are merged together.

7.6 Completion & negotiation

Philips did not commit to buying the product when the test case would have a positive result. After the test case, Metis was considered as a very good tool for Philips, but Computas is not proposed yet and perhaps they will never be. The situation now is that Philips has not decided yet to use a new EA tool. A benefit of this selection process is that if in the future a certain tool is required then there is already good information available. That information can serve as a starting point in a new tool selection process. The test case held with Metis is the same as a Proof of Concept (PoC), with the exception that after the tool was considered as satisfactory no immediate proposition followed.

Thus this phase is not a part of this selection process. Nevertheless this phase is inevitable in any complete selection process. If Philips decides to adopt an EA tool, it has to complete the selection process with this phase. Therefore the important aspects of this phase will be mentioned in this section. The objective of this last phase is to verify the principle choice and to make all the final decisions concerning the completion of the selection process. This phase should result in [De Heij]:

- A (functional) confirmation of the (in principle) chosen candidate from the short list, as far as there was any doubt left of to the previous phase;
- An impression of the software modifications required and the consequences for the organization;
- An implementation plan based on the above, including the responsibilities and tasks (and those of the supplier), the phasing and timing of all activities and the project organization;
- A contractual agreement composed of all of the elements listed above;
- The good support from within the organization for the contracted EA tool as well as for the implementation plan.

However, the quality of the final result (the right choice among other things) is hard to measure and is usually not measured at all. This means that mistakes in the selection process are only found out at a very late stage or remain undetected. A system is chosen and a contract is concluded but the mistakes made (or opportunities missed) in the earlier stages only become clear during the implementation or when the system is operational. Often they never become visible, but the result is an overly expensive or delayed result. Therefore this selection process was done very carefully and a lot of time was spent in collecting the right information [De Heij].

7.7 Selection process based on data structures

In the book of Jos C.J. de Heij also a selection process of Business Information Systems is described, which is based on analyzing data structures. Data structures are a necessary precondition, but not enough to indicate a certain functionality. Only a lack of a certain data structure indicates that some functionality is not possible. A tool can only provide a certain functionality if it uses the right data structure, but the right data structure does not guarantee that the tool provides a certain functionality. Therefore a traditional test with the EA tools is needed to find out if the EA tool provides all the needed functionality. This test should then only be held with the EA tools that contain the right data structures. Assessment of data structures can contribute in this way in an EA tool selection process.

In the case of the selection process of an EA tool, the assessment of data structures can indicate if certain functionality can or cannot be provided. In Metis for example the used data structures (templates) are not fixed and the user is able to make changes in it. But other tools, like ARIS, contain a fixed meta-model. Changes in these data structures can only be done by the vendor.

Assessment of data structures is only profitable when software selections are carried out very frequently. The investment needed for the direct assessment of the data structures of EA tools is significantly lower. Because the selecting process of an EA tool will not take place frequently, tool selection based on data structures is not advisable for Philips. Therefore the selection process based on data structures was not considered in this selection process [De Heij].

8 Conclusions and Recommendations

This chapter is split up in three sections. Section 8.1 contains the conclusions about the theory. Section 8.2 contains the conclusions about the selection process. The last section will contain an overall conclusion and the recommendations for the tool selection process and using an Enterprise Architecture at Philips.

8.1 Conclusions about the theory

Current Enterprise Architecture tools are only capable of describing the architecture, considering different views and relationships, and provide limited analysis. They are not related to operational processes at all. In section 3.7 a view about the future was given, in which Enterprise Architecture takes an important role in (operational) management. Enterprise Architecture tools and frameworks are slowly developing towards these future prospects. In the meantime there are a lot of uncertainties to use the current tools and frameworks in a proper way. In this thesis there were identified three approaches of Enterprise Architecture. Now Enterprise Architecture is in many times seen as the responsibility of the IT department. Enterprises have to discover the other possibilities of Enterprise Architecture, so they can deploy Enterprise Architecture also in other situations.

The frameworks are still too much focused on describing the enterprise and Business - IT alignment. In the future other frameworks will appear to support also other approaches of Enterprise Architecture. The formal description of the IAF framework with the objects and relationships from the test case showed that the current implementation was not fully compliant with the IAF framework. The formal description is a good way to have a clear overview of what is allowed and what is not and this is a good starting point for the Enterprise Architecture tools.

8.2 Conclusions about the Enterprise Architecture tool selection for Philips

Philips needs an Enterprise Architecture tool that helps them to represent their various architectural descriptions on a comparable basis and support the definition of mutual relationships between enterprise level and divisional level architectures. Because of the introduction of the IAF framework, this framework has to be supported in a structural way. An Enterprise Architecture tool is the solution for this. The main objective of the Enterprise Architecture within Philips is Business - IT alignment. Communication with all the stakeholders (on corporate level as well as on divisional level) is also an important benefit of the tool. Another requirement is that the tool provide in 'traceability of arguments': to make it clear and understandable why a certain decision is taken. Finally the tool should be able to support the IAF framework and allow customization and adaptation of the repository meta-model. With the help of Enterprise Architecture Philips has to be able to make better decisions in the field of IT. Because of the use of Enterprise Architecture the Babel-like confusion will be history. Now people have different views or opinions of the current situation. Not everybody understands the reality and based on that the wrong decisions can be taken. With Enterprise Architecture there is clearness about the reality and this results in a better IT governance.

In chapter 7.5 the test case with Metis was described. The most important results of this test case were:

- **Metis:** Metis is a very nice tool to work with and during the test case everything that was necessary for the test case could be created. The graphics are important in Metis, every model can be drawn exactly as the users are used to;
- **Test criteria:** the most important test criteria could be easily implemented;
- **Results on Hydra and PHERA:** Hydra and PHERA are not consistent, Hydra is not complete, the Hydra definitions are not strict enough and by using the IAF framework relevant business questions could be generated;
- **Contribution of Metis for Philips:** Metis is very suitable for Philips because it can provide cooperation between the different PD's and the Corporate level. It can be an important help to move 'towards one Philips'. Metis supports the goals of Corporate IT in finding synergy opportunities. Without the help of a tool that would not be feasible because there is too much information and the patterns are too complex for human beings.

After the first steps in the selection process Metis was considered as the most promising tool for Philips. After the test case this consideration is still valid. The selection process is now in a final stage at Corporate IT. Due to another view of Enterprise Architecture at the management of Corporate IT there is no interest to continue the tool selection process by proposing Computas. Therefore the selection process is currently on hold within Corporate IT. Besides Corporate IT, product divisions can decide whether to continue with the tool selection process, to replace the tools they currently use, or even to select other tools and start the tool selection process again.

8.3 General Conclusions and Recommendations

Philips needs an Enterprise Architecture tool for a better communication and cooperation between its divisional levels and corporate level in the field of architecture. The cooperation is mainly focused on cooperation in the field of IT, such as: interoperability, shared solutions, shared development, etc. This strategy is also known as moving 'towards one Philips'. That strategy will make Enterprise Architecture even more relevant, because there will be an increased focus on sharing.

Now some recommendations to continue this selection process will be given. These recommendations for Philips, with which these thesis ends are:

- Consider using Metis instead of other architecture tools. Metis is a suitable Enterprise Architecture tool. Consider the tools, which are currently used in Philips and check if Metis can be used successfully in combination with these current tools or that it can substitute them. But during the test case a team server could not be used. To test if Metis is the right tool for cooperation between different architects, who are working at the same models, another test is required. In that test cooperation, version control and authorization have to be closely examined;
- Consider Metis every time architecture is involved. Metis has to be considered as a possible tool to provide the architectural descriptions, like projects between different PD's. For instance, the projects that are a result of the 'towards one Philips' strategy can

be supported with a tool like Metis. In this tool the different views of every PD can be easily generated;

- Use Metis to support the IAF framework. Because using the IAF framework is decided, architects have to find a way to use the IAF framework in a structured way. Metis can provide in using the IAF framework in a structured way. Therefore architects and managers have to get acquainted with Metis and to then to decide to use and implement it. If Metis becomes the standard in Philips, cooperation in the field of architecture organized with the IAF framework will be possible;
- Start the introduction of Metis in a small way. To have success with Enterprise Architecture projects and Metis there has to be generated a maturity and willingness among the employees to use it. Architects and managers have to agree that they need an Enterprise Architecture tool and that it is of a benefit for them. Since there is not such maturity and willingness yet, introduction of a tool will not be beneficial. This requires a careful introduction of Enterprise Architecture and Enterprise Architecture tools. Another benefit of starting to use Metis in a small number of projects that there will be generated architectural descriptions of different parts. Architects are able to learn from each other to reuse existing architectural descriptions in other projects. In the first stage Metis can be used to describe the architectures, but in a later stage, when maturity is increasing and people consider Enterprise Architecture as useful, Metis can also be used for deploying searches, simulations, etc. Metis is a tool that is very suitable for extensive analysing of architectures. Therefore the work that is done by using Metis in small projects can be used in a later stage when there is more required of the architecture in decision-making. When in the future Metis can support more functionalities as described in section 3.7 there are already good starting points available.

In conclusion Metis should be considered as a suitable tool, which can be used in every architecture process. Product divisions have to choose their own strategy with regard to architecture. If they are considering architecture tools, Metis should be put on their short list. The introduction of an Enterprise Architecture tool should be done with a lot of care. It is better to start in a small way and after success the use of Metis can be spread out. The created models are the basis of the growing number of models and the acquired knowledge can be used in other projects and divisions. At the moment the divisions Semiconductors and Consumer Electronics are showing interest in Metis and perhaps these divisions can take the lead in adopting Enterprise Architecture in Philips. Therefore this quotation reflects in this situation, with reaching Peking as a metaphor for the full implementation and use of an Enterprise Architecture tool in the whole organisation of Philips:

'A walking-tour to Peking also starts with the first steps, if you do not make them you will never get there'

[Wim Kan]

9 Bibliography

[Ambler et al.] Scott W. Ambler, James McGovern, Mike Stevens, James Linn, Vikas Sharan, and Elias Jo, *The Practical Guide to Enterprise Architecture*, chapter 7, Prentice Hall, 2003, www.agiledata.org/essays/enterpriseArchitecture.html, consulted on 28 July 2004.

[Archimate] Archimate, *State of the Art in Architecture Frameworks and Tools*, https://doc.telin.nl/dscgi/ds.py/Get/File-22327/archsota.pdf, consulted on 8 June 2004.

[AWG1998] Architecture Working Group, *Draft Recommended Practice for Architectural Description IEEE P1471/D4.1*, www.informatik.hu-berlin.de/~holz/Vorlesung/VS/p1471_v41.pdf, consulted on 22 July 2004.

[BPT2003] Business Process Trends: *Developing an Enterprise Architecture*, Whitepaper, January 2003.

[BPT2004] Business Process Trends: *Enterprise Architecture,* Newsletter, Volume 2, No. 1, January 2004.

[BPTrends] Business Process Trends spotlight, volume 1, number 3, www.bptrends.com/publicationfiles/bptspotlight0518.pdf, consulted on 16 August 2004.

[Bredemeyer] Bredemeyer Consulting, *Enterprise Architecture and Related Definitions*, www.ewita.com/EA_Overview/Definitions/ArchitectureDefinitions.htm, consulted on 9 August 2004.

[Bruegge & Dutoit, 2000] Bernd Bruegge and Allen H. Dutoit, *Object-Oriented Software Engineering: Conquering Complex and Changing Systems,* Prentice Hall, 2000.

[Bruijne et al.2004] A.J. de Bruijne and L.H.M. Osse, *Inbedding van architectuur in organisaties*, www.serc.nl/lac/2003/papers/A.deBruijne.pdf, consulted on 31 March 2004.

[Cap] Cap Gemini: *IAF Essentials*, workshop, Evoluon Eindhoven, 30 June and 1 July 2004.

[CapGemini] CapGemini, www.nl.capgemini.com/innovatie/corporatetransparency/index1.htm, consulted on 9 June 2004.

[CapGemini2] CapGemini, www.nl.capgemini.com/diensten/architectureservices/visie.htm, consulted on 5 August 2004.

[CapGemini3] CapGemini, http://opleidingen.capgemini.nl/df.html?url=/opleidingen2/cluster/index.iphtml?wgid=178&taal=nl, consulted on 12 August 2004.

[CDC] United States Department of Health and Human Services Centers for Disease Control and Prevention, www.cdc.gov/irmo/ea/ea.htm, consulted on 9 August 2004.

[De Heij] Jos C.J. de Heij, *Assessment of Business Information Systems by Data Structures*, Kluwer Bedrijfswetenschappen, 1996.

[Delen et al.] D. Delen, P.C. Benjamin: *Towards a truly integrated enterprise modeling and analysis* *environment*, Computers in Industry 51 (2003) pp. 257–268, Elsevier.

[Dimensie] Studievereniging Dimensie Universiteit Twente, *Faculteit der Gedragswetenschappen*, www.svdimensie.nl/ samenvattingen/ samenvatting%20plomp.doc, consulted on 24 August 2004.

[DIS] Department of information Services, www.dis.wa.gov/ academy/ presentations/ 00_12_20Transformation.ppt, consulted on 1 April 2004.

[DMReview2004] DMReview, *Beyond the Data Warehouse: Do You Need an Enterprise Information Architecture?* www.dmreview.com/ editorial/ dmreview/ print_action.cfm?articleId=1005222, consulted on 9 August 2004.

[Ernst & Young] Ernst & Young, *Nieuws: Nieuwe IFRS-regels dwingen een hogere kwaliteit directieverslag af*, www.ey.nl/ ?pag=2106&nieuws_id=1996, consulted on 3 August 2004.

[FAA1998] Federal Aviation Administration, *Federal Information Architecture Initiatives*, February 1998, www.faa.gov/ niac/ pdf/ wn18_fia.pdf, consulted on 23 June 2004.

[FAQ] Frequently Asked Questions from the website for IEEE Std 1471, www.pithecanthropus.com/ ~awg/ public_html/ ieee-1471-faq.html, consulted on 12 July 2004.

[Forrester2004] Forrester Consulting, *Tool Choice Recommendations Targeting the Combination of Enterprise and Portfolio Mgt*, June, 30, 2004, http:/ / weat.wi.gov/ docview.asp?docid=698, consulted on 21 September 2004.

[GEAO] GEAO, Enterprise Architecture Tool Strategy, The Enterprise Architecture Conference - Wellington 4-6 Nov 2003, www.geao.org/ pdf/ ea_tool_strategy_v1_extract.pdf, consulted on 14 July 2004.

[GeoNieuws] GeoNieuws, *Architectuur binnen RWS, een stuurmiddel bij uitstek*, www.geonieuws.nl/ html/ pdf/ geo-2004-1/ o-05-04-1.pdf, consulted on 10 June 2004.

[Gharajedaghi] Jamshid Gharajedaghi and Rusell L. Ackoff, *Mechanisms, Organisms and Social Systems*, www.interactdesign.com/ mech.html, consulted on 16 August 2004.

[Goedvolk] Hans Goedvolk, *Integrated Architecture Framework*, http:/ / home.hetnet.nl/ ~daan.rijsenbrij/ arch/ publ/ White%20paper%20IAF%20v1.0.ppt, consulted on 3 November 2004.

[Gray2002] Elizabeth T. Gray, *Outsourcing Relationship Management: The Business Case and a Self Test*, www.alliancemanagementpartners.com/ news/ article_1.htm, consulted on 11 August 2004.

[Greefhorst et al.] Danny Greefhorst en Henk Koning, *Dimensies in architectuur*, http:/ / www.cibit.nl/ site.nsf/ 0/ 5BD279213AE3539BC1256E37005348E6/ $file/ LAC2002DimensiesinArchitectuur.pdf, consulted on 3 November 2004.

[Greefhorst] Danny Greefhorst, *Software-architectuur in vogelvlucht*, www.array.nl/ release/ artikelen/ 2003/ 2_Greefhorst_architectuur.pdf, consulted on 17 December 2004.

[Hilliard] Rich Hilliard, *IEEE-Std-1471-2000 IEEE-Std-1471-2000: Recommended Practice for Recommended Practice for Architectural Description of Architectural Description of Software-Intensive Systems*, www.enterprise-architecture.info/ Images/ Documents/ IEEE%201471-2000.pdf, consulted on 12 July 2004.

[Hoogervorst] Dr. Ir. J.A.P. Hoogervorst, *Enterprise Architecture: Enabling Integration, Agility and Change*, www.serc.nl/ lac/ 2003/ academic_papers/ J.Hoogervorst.pdf, consulted on 15 April 2004.

[IEAC] The International Enterprise Architecture Center, www.ieac.org/ about/ main.html, consulted on 10 August 2004.

[IEEE] IEEE Computer Society: *IEEE Standard 1471-2000: IEEE* Recommended Practice for Architectural Description of Software-*Intensive Systems, 2000.* www.ieee.org

[IFEAD] The website of the *Institute for Enterprise Architecture Developments*, www.enterprise-architecture.info, consulted on 2 March 2004.

[Kazman et al.] Rick Kazman, Len Bass and Paul Clements, Software Architecture in Practice (2nd edition), Addison Wesley, 2003.

[Kersten et al.] Prof. dr. H.M.P. Kersten and Prof. dr. C. Verhoef, *Door bankiersogen kijken naar IT*, www.math.vu.nl/ ~hkersten/ pdf/ artikel_banking_review_ITportfoliomaart2003.pdf , consulted on 9 June 2004.

[KUN] KU Nijmegen, *Architectuur in de Digitale Wereld*, www.architectuur-in-de-digitale-wereld.nl, consulted on 16 March 2004.

[Land] Rikard Land, *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, www.idt.mdh.se/ phd/ courses/ sa/ IEEE_RecommendedPractice.pdf, consulted on12 July 2004.

[Linz] Peter Linz, *An introduction to formal languages and automata*, Sadbury, Massachusetts: Jones and Bartlett publishers, 1997

[Macauley] Andrew Macauley (CGEY), *Enterprise Architecture Design and the Integrated Architecture Framework*, Microsoft Architects Journal EMEA edition; issue 1, January 2004. http:/ / msdn.microsoft.com/ library/ default.asp?url=/ library/ en-us/ dnmaj/ html/ aj1entarch.asp, consulted on 3 June 2004.

[McGriff] Steven J. McGriff, *Knowledge Base / Systems Theory*, www.personal.psu.edu/ faculty/ s/ j/ sjm256/ portfolio/ kbase/ Systems&Change/ systems.html, consulted on 19 August 2004.

[OCP] The website of *Organisation Consulting Partnership*, www.ocp.co.uk/ busarch_content.htm, consulted on 15 April 2004.

[OOPSLA'96] William E. McCarthy, Julia Smith David,Brian S. Sommer, *The Evolution of Enterprise Information Systems -- From Sticks and Jars Past Journals and Ledgers Toward Interorganizational Webs of Business Objects and Beyond*, http:/ / jeffsutherland.org/ oopsla96/ mccarthy.html, consulted on 8 February 2005.

[Peyret2004] Henry Peyret, *Getting Value From Enterprise Architecture Tools*, August 9, 2004.

[Proforma] Proforma Corporation, *Successful Enterprise Architecture*, whitepaper, www.proformacorp.com/ downloads/ wp-ProformaEnterpriseArchitecture.pdf, consulted on 24 March 2004.

[Project Auditors] Project Auditors, www.projectauditors.com/ Dictionary/ G.html, consulted on 30 August 2004.

[Rietveld et al. 2004 ITG 1] Mrs. Ir W.F. Rietveld and dr ir ing A.P. Nagel, *The sheets of the course IT-governance given at the TU/e*, week 1, spring 2004.

[Rietveld et al. 2004 ITG 4] Mrs. Ir W.F. Rietveld and dr ir ing A.P. Nagel, *The sheets of the course IT-governance given at the TU/e*, week 4, spring 2004.

[Rietveld et al. 2004 ITG 6] Mrs. Ir W.F. Rietveld and dr ir ing A.P. Nagel, *The sheets of the course IT-governance given at the TU/e*, week 6, spring 2004.

[Rietveld et al.] Erica Rietveld and Niels Klinkenberg, *What is Business Architecture?*, Grexx white paper, version 0.3, April 2004.

[Rijsenbrij et al.] Daan Rijsenbrij, Jaap Schekkerman and Harry Hendrickx, *Architectuur, besturingsinstrument voor adaptieve organisaties*, Uitgeverij Lemma BV, Utrecht, 2002.

[Rogier1998] Jacobus Johannes Hermanus Rogier, *De wisselwerking tussen organisatie en markt*, www.ub.rug.nl/ eldoc/ dis/ management/ j.j.h.rogier/ h6.pdf, consulted on 18 August 2004.

[RRB] Railroad Retirement Board, Bureau of Information Services, www.rrb.gov/ BISPublic/ bisarchitecture.htm, consulted on 1 April 2004.

[Schekkerman] Jaap Schekkerman's speech at the Enterprise Architecture Practioners Conference, 20 – 22 April 2004, Belgium, www.enterprise-architecture.info/ Images/ Documents/ Comparative_Survey_of_EA_Frameworks.pps

[Schekkerman1999] Ing. Jaap Schekkerman, *The Architecture Process Cycle: The Architect; The IAF WinWin Spiral Model*, www.serc.nl/ lac/ LAC-2001/ lac-1999/ docs/ schekkerman.doc, consulted on 5 August 2004.

[Schekkerman2003] Jaap Schekkerman, *How to survive in the jungle of Enterprise Architecture Frameworks: Creating or choosing an Enterprise Architecture Framework*, Victoria: Trafford, 2003.

[Schellen] Gé Schellen, *Enterprise Architecture Program Bottom up stream: Corporate IT Architecture*, document written for Philips.

[SPC] Software Productivity Consortium, www.software.org/ pub/ architecture/ eap.asp, consulted on 10 August 2004.

[Texas] The Texas State Library and Archives Commission, www.tsl.state.tx.us/ ld/ pubs/ compsecurity/ glossary.html, consulted on 11 August 2004.

[Texas2003] State of Texas, 2003 State Strategic Plan for Information Resources Management, www.dir.state.tx.us/ pubs/ ssp2003/ ssp2003.pdf, consulted on 10 August 2004.

[UML] Object Management Group, Unified Modeling Language Specification, version 1.1, OMG AD/ 97-08-05, 1997.

[USGAO] USGAO: *Information Technology, Leadership Remains Key to Agencies Making Progress on Enterprise Architecture Efforts,* United States General Accounting Office, GAO-04-40, November 2003, www.fpmi.com/lerpress/d0440.pdf.

[Van Giessel] Martijn van Giessel, Master's thesis: *Process Mining in SAP R/3, A method for applying process mining to SAP R/3.*

[Van Hee] Prof. dr. K.M. van Hee, *"ICT-governance and the role of Architecture"*

[Van Hee1994] K.M. van Hee, *Information Systems Engineering, A formal approach*, Cambridge University Press, 1994

[VanHee2002] Kees M. van Hee and Wil M.P. van der Aalst, *Workflow management: models, methods, and systems*, London: MIT Press, 2002.

[Vliet et al.] Hans van Vliet, Danny Greefhorst and Henk Koning, *De dimensies in architectuurbeschrijvingen*, www.cs.vu.nl/~hans/publications/y2003/dimensies.pdf, consulted on 4 August 2004.

[Wikipedia] Wikipedia, http://nl.wikipedia.org/wiki/SAP_(bedrijf), consulted on 5 January 2005.

[Zachman] John A. Zachman, *Packages Don't Let You Off the Hook!*, www.frameworksoft.com/html/body_packages.html, consulted on 05 October 2004.

[Zachman1987] J. Zachman, *A Framework for Information Systems Architecture*, the IBM Systems Journal (Vol. 26, No. 3, 1987), www.va.gov/oirm/architecture/EA/theory/zachman.pdf, consulted on 15 June 2004.

[ZIFA] The Zachman Institute for Framework Advancement, www.zifa.com, consulted on 18 March 2004.

Appendices

Abbreviations

AD	Architectural Description
BA	Business Architecture
BPM	Business Process Modelling
BPR	Business Process Redesign
CIO	Chief Information Officer
EA	Enterprise Architecture
EAP	Enterprise Architecture Planning
EIS	Enterprise Information System
ERP	Enterprise Resource Planning
EU	European Union
IAF	Integrated Architecture Framework
IEEE	The Institute of Electrical and Electronics Engineers, Inc.
IEEE 1471	The IEEE Standard 1471-2000: IEEE Recommended Practice for Architectural Descriptions of Software Intensive Systems
IFRS	International Financial Reporting Standards
IT	Information Technology
PD	Product Division
PHERA	PHilips Enterprise Reference Architecture
PoC	Proof of Concept
RFI	Request For Information
SC	Semiconductors
SITAR	SITe ARchitecture
UML	Unified Modeling Language

List of figures

List of tables