Martin Op 't Land, Erik Proper,
Maarten Waage, Jeroen Cloo and
Claudia Steghuis

Enterprise Architecture

Creating Value by Informed Governance

January 29, 2014

Springer

Foreword by Frank Baldinger

I feel especially pleased and honoured to write the foreword of this book because it is the first book supporting a Master's curriculum on EA (Enterprise Architecture) at Radboud University Nijmegen, the Netherlands. The book is also written for practitioners of EA in end-user organisations, and for IT-suppliers. The book focuses on the needs underlying enterprise architecture. The needs from a business perspective are considered, as well as the needs from the perspective of designers, engineers, and developers.

As you can read in Chapter 3, there are many definitions of "Enterprise Architecture". The authors of this book state: *The variety in these definitions does seem to indicate that the field of enterprise architecture is still in its infancy*. I would rather contribute this variety to growth, evolution, and enrichment within the EA domain, and to the tendency to define EA from the points of view of a variety of stakeholders. Many people (including myself) do still feel an urgent need for a converging EA-discipline, including international standards and acceptance. Sadly there have only been some weak signs of such a development up to now. Simply put: in my opinion EA has now evolved from infancy to adolescence.

Twenty years after the first publications and books on Enterprise Architecture, the domain is evolving slowly, but in a straight line, towards a more business driven approach. It has been transformed from process and technology driven to more business and information driven. In the last few years, it has also evolved from a framework-driven approach to a business-capability approach. This implies that the need for competencies with respect to information science and system engineering is extended with the need for "other" competencies, for example focusing on business stakeholder management or organisational problem diagnostics. As a result the activities and competencies of an enterprise architect are increasingly heterogeneous and evolve at least in two directions: towards an Architecture Engineering discipline and towards a discipline where Enterprise Architects work more closely to the Business.

The authors of this book are lecturers involved in the Master's curriculum on enterprise architecture taught in tandem by Capgemini and the Radboud University

Nijmegen. They have a broad and deep experience in both disciplines involved, rooted in their daily work at Capgemini and the University. The resulting chemistry between competencies and mindsets gives a very special flavour to this book: a flavour of enrichment of both sets of competencies. There is, however, also a built-in tension: the tension between the mindsets of the people working in the domains of business engineering versus system/information engineering. As both aspects are integrated in this book in a very natural way, the result has considerable added value for practical use in organisations, where the same tension exists. Therefore, this approach promises to be very useful to both students and experienced enterprise architects working in either domain.

As the Master's course on Enterprise Architecture at Radboud University is continuously evolving, this book shows that it has certainly the capability to support the need for convergence on EA and help the international EA-community to strive in that direction. This is the common ground for both the Master's curriculum on EA and the Dutch National Architecture Forum (NAF). The NAF community platform for EA is used by several authors of this book to validate and extend their ideas on EA. The NAF has about 70 organisations as members, distributed over three pillars: user-organisations, suppliers (including consultancy organisations) and scientific institutions (including educational institutions). They work together in working groups on mutually interesting EA-subjects, enriching the output with their distinctive views on the world of EA.

Many words have been written on the subject of EA, but I believe this book will stand the test of time and be recognised as a foremost contributor to the evolving body of knowledge of EA.

Ir. A.F. (Frank) Baldinger
Chairman NAF (Netherlands Architecture Forum)

Foreword by Frank Harmsen

When I first read a draft version of the manuscript of this book, I was proud. Proud on the authors, who spent many, many hours passionately writing one Chapter after another. Writing a book about a maturing competence such as Enterprise Architecture is difficult. The lack of good definitions and the many, many homonyms and synonyms roaming around in the area is one hurdle. Some concepts are still in a stage that they require further research. Some are defined in theory, while in need of much more empirical testing. And, last but not least, as in every scientific discipline, there are various 'schools' advocating different paradigms and ways of thinking. Nevertheless, the authors have managed to produce a nice synthesis and overview of the domain. I congratulate them with this!

I was also proud on the collaboration between Academia and 'the industry'. For a company like Capgemini, such a collaboration is of vital importance. Capgemini continuously needs to innovate in order to help our clients in the best possible way. And we cannot do that on our own. I personally believe that the connection between Academia and the consulting industry can be much more stronger. There is so much low hanging fruit to be captured if scientists and consultant co-operate, but also on strategic issues requiring years of research, the universities and the consulting firms need to find each other. This is not only important to them, but to society in general. This book is a good example of how Academia, in this case the Radboud University Nijmegen, and the industry can jointly create an outcome that can help our clients to cope with the challenges and opportunities they face.

These challenges and opportunities are manifold. We live in an age in which change is the only constant. Complexity is omnipresent and so is the need to be agile and to transform. Businesses, i.e. our clients, need instruments to transform while their operations need to continue and improve. It is like redesigning and building a rocket that is already on its way to Mars - the term rocket science fully applies to the transformation challenges of our clients! And yet, these instruments are already available. One of the more important ones is Architecture, and in particular Enterprise Architecture. Enterprise Architecture is the instrument that helps organisations

in analysing and structuring their current complexity, and in designing and managing their continuous transformation process to become agile, efficient and effective.

The mission of Capgemini is: 'Enabling Transformation'. It is therefore not surprising that Enterprise Architecture is in the heart of our company. We consider it of strategic importance. We have a long standing tradition that started in the eighties and continued in the nineties with the advent of our Integrated Architecture Framework (IAF), which is currently incorporated in The Open Group Architecture Framework (TOGAF). This tradition has been vital to us and beneficiary to our clients as well. This book is part of this tradition, and I thank the authors for writing it.

Prof.dr. A.F. (Frank) Harmsen
Manager Service Line Architecture, Capgemini Netherlands

Preface

This book is positioned as a first in a series of books on enterprise architecture needed for a *Master of Enterprise Architecture* program, and is targeted both at university students and practitioners with a drive to increase their understanding of these fields.

As an introductory book, this book aims to explore the concept of enterprise architecture. At first glance, writing such an introductory book might seem as a straight forward task of setting up a structure and filling in 'the blanks'. However, writing this book turned out to be a pleasant journey of discovery. Based on our past experiences, each of us had a clear understanding of enterprise architecture, based on several years of experience and insight in the field. However, when we started writing this book, and each of us exposed our individual understandings, it became apparent that our understanding of the field differed in several ways. This prompted several discussions leading to an abundance of new insights. Without exception, these discussions took place in a pleasant and open atmosphere, fuelled by our shared drive for understanding and increased insight. We are now even more convinced than before, that the field enterprise architecture is a true multi-disciplinary profession.

In the resulting book, we would like to share our insights, while also hoping to continue our discussions, now also involving *you* as a reader. We also realise that the journey is still far from complete. While this introductory book provides an overview of the field of enterprise architecture from the perspective of our insights, many aspects need further refinement. In our opinion this also applies to the field as a whole. As we will also conclude in this book, the field needs more maturing, and in writing this book we hope to provide a humble addition to this maturation process.

Martin Op 't Land, Erik Proper, Maarten Waage, Jeroen Cloo and Claudia Steghuis
Utrecht
The Netherlands

Acknowledgements

The authors of this book are not the only ones who have contributed toward the creation of this book. Firstly, we would like to thank Cornelly Spier for doing the project management of this book. Without her enduring support, this book would not have been finished.

In writing this book, two much needed bridges needed to be build. One between academia and industry, and one between business and IT. It is our belief that the further development of the field of enterprise architecture requires a strong interaction across these bridges. While sharing this vision, our sponsors provided us with the support needed to create this book. We would therefore like to thank Capgemini's architecture service line, in particular the architecture practices of the financial services, public and products sectors, as well as the Institute for Computing and Information Sciences from Radboud University Nijmegen for their support in creating this book.

Furthermore, we would like to thank the reviewers of earlier versions of this book's manuscript: Jan Achterbergh, Piet Adriaanse, Frank Baldinger, Dave van Gelder, Bas van Gils, André Gronsveld, Frank Harmsen, Wijke ten Harmsen van der Beek – Hamer, Herman Hartman, Jan Hoogervorst, Inderjit Kalsi, Mendel Koerts, Marc Lankhorst, Mieke Mahakena, Ronald Orlemans, Lucas Osse, Arnold van Overeem, Bart and Michal Papegaaij, Bert Sneep and Anton van Weel for their valuable contributions.

Contents

Chapter 1
Introduction

1.1 Background

This book has been created in an effort to develop a textbook for one of the key courses of a *Master of Enterprise Architecture* program. It is a first in a series of books needed to further underpin this Master's program with textbooks combining a sound theoretical base with practical insights, and has been authored in a close collaboration between industry and academia. In authoring this book, we have been driven primarily by the need for textbooks for the further professionalisation of enterprise architects as well as education of students aspiring to become enterprise architects. As such, the books needed for the *Master of Enterprise Architecture* program, will be targeted both at university students, as well as practitioners with a keen interest in gaining a thorough understanding of these fields.

In this book, we explore the concept of enterprise architecture. An enterprise is understood as comprising of at least business, human and technological aspects. To be more precise, we define enterprise as *a goal oriented cooperative to be implemented by people and means*. In creating, evolving and/or transforming enterprises, several challenges come to the fore on how to govern such changes. Enterprise architecture is an emerging means of governing these changes. The key drivers for this means therefore are the enabling of informed decision making on these changes, as well as ensuring compliance to these decisions.

This book aims to provide an overview of enterprise architecture including the process of creating, applying and maintaining it, while taking a fundamental view on the field of enterprise architecture. In doing this, we aspire to create an understanding of the mechanisms underlying enterprise architecture, as well of its role as a governance and decision making instrument bridging the gap between an enterprise's vision, strategy and change projects. This role is also taken as a starting point to explore the results that may be produced as part of an enterprise architecture, the process in which these are to be produced, and the role the architect will play in this process. As such, this book does not describe a specific method to develop an enterprise (IT) architecture [21, 35, 148], nor does it define a specific modelling language

for enterprise architecture [20, 78] or does it subscribe to a specific enterprise architecture framework [30, 45, 139, 154, 155]. As mentioned above, it rather aspires to offer the reader a fundamental way of thinking on enterprise architecture. The field of enterprise architecture still seems rather immature. While this book aspires to take a more fundamental view, we will quite regularly run into situations where insight from practitioners seems to make certain indications about, for example, the potential role/value of enterprise architecture, while scientific evidence is lacking. In this book we also not provide this much needed underpinning. This remains left as challenges to the scientific community. Such challenges will also be summarised in the final Chapter, where we list a range of research challenges that need to be addressed when maturing the field.

1.2 Outline of the book

In Chapter 2 we start with an overall exploration of the motivations why enterprises turn to *enterprise architecture* to aid them in meeting modern day challenges. Developments such as globalisation, the fusion of business and IT, new technologies, the introduction of new business models and new regulations, occur at a higher pace than ever. This requires modern day enterprises to be able to adapt themselves swiftly to these changes. This puts a challenge on managers to make the right decisions at the right time for both short and longer term needs. The increasing complexity of the issues involved, as well as the growing diversity and heterogeneity of the concerns and stakes of the stakeholders involved, render pre-existing approaches less adequate. This calls for a new governance instrument, a call that is to be answered by the instrument of enterprise architecture. Chapter 3, therefore continues by discussing enterprise architecture as a means to meet the needs discussed in Chapter 2. It provides a historical perspective on enterprise architecture, followed by a discussion on the governance paradigm which will be used to underpin our definition of enterprise architecture. In addition to providing the definition of architecture as it will be used in this book, the core concepts of enterprise architecture will be discussed.

Equipped with this understanding, Chapter 4 continues with a discussion of the results that can be produced when architecting an enterprise. In discussing these results we will distinguish several dimensions along which to classify and position them. Among these dimensions, we will distinguish between:

Subject dimensions – Dealing with the classification of the subject, relative to the enterprise being architected, with which the result is concerned (e.g. business, application, enterprise-wide, system specific, contextual, conceptual, logical, etc).

Purpose dimensions – Expressing the purposes for which the result is intended (e.g. analytical, collaborative, informative, decisive, etc).

Form dimensions – Concerned with the forms in which the result may occur (e.g. principles, patterns, graphical models, formal models, textual descriptions, informal sketches, implicit knowledge, attitudes, etc).

These dimensions will give rise to the so-called architecture frameworks, such as Zachman [155], TOGAF [139], RM-ODP [64], DYA [147] and IAF [30, 45]. The purpose dimension is elaborated in views and viewpoints for specific stakeholders [60].

In Chapter 5, we zoom in on the processes involved in creating, applying and maintaining enterprise architecture, covering such activities as:

- joint conceptualisation of problems, strategies or solutions,
- risk assessment and mitigation,
- decision making,
- assessing alternatives,
- transformation planning,
- offering guidance to development projects and
- ensuring compliance of development projects.

In addition to activities, as exemplified above, in which the process of architecting is *act*ed out, we also discern *plan*ning, *learn*ing and *organis*ing activities. The planning activities involve the deliberate planning of which activities to undertake in the architecting process. Enterprise architecting is a continuous process involving the creation, modification, enforcement, application and dissemination of different results. This continuous process should be in sync with developments in the environment of the enterprise as well as developments internal to the enterprise, including both its strategy and its operational processes. We also stress the fact that there is no one-size-fits-all approach to architecting, and that a situational approach is needed. We will identify different approaches to architecting [119, 139, 148], but refrain from casting judgment on the relative quality of these approaches. Especially since enterprise architecting as a professional field is still rather in its infancy, there is a need for continuous *learn*ing. In other words, the activities involved in enterprise architecting should be scrutinized on their efficiency and effectiveness, and where possible, lessons learned should be recorded and taken into account in future situations. Combining the *act*ing, *plan*ning, and *learn*ing activities leads to a *plan-act-learn* cycle. In order to get this *plan-act-learn* cycle operational, and keep it operational, an explicit architecture function must be implemented in the enterprise (*organise*).

As a next step, we turn our attention to the professionals who are responsible for the execution of the activities involved in enterprise architecting: *the architects*. Chapter 6 therefore focuses on the responsibilities of architects and the desired competencies. Based on some studies into the skills of architects [26, 90, 139], the *Architecture Skills Framework* from TOGAF [139], a survey conducted among Capgemini enterprise architects, as well as our experiences in teaching future enterprise architects, we will discuss the essential competencies that should ideally be exhibited by an architect and the responsibilities they should be willing to accept in doing their work.

Given the demanding needs on enterprise architecture, the discussions provided in this book, as well as day-to-day practices of enterprise architecting, one can only conclude that the field of enterprise architecting is far from mature. As a profession we are not yet able to aid organisations in solving their transformation problems in a repeatable and predictable fashion. To remedy this, several aspects of our field need further elaboration and even fundamental research. Before concluding this book, Chapter 7 therefore discusses several research challenges that remain.

At the end of the ensuing Chapters we will include some *discussion statements*. These statements are by no means intended to be true or false, but rather aim to spark discussion. With these statements, we invite the readers of this book to reflect on these statements, in order to sharpen their opinion of, and understanding about, enterprise architecture.

Chapter 2
The Need for a New Instrument

2.1 Introduction

As a result of developments such as globalisation, the fusion of business and IT, the introduction of new technologies, novel business models, etcetera, enterprises are confronted with an increasing variety of options to deal with an ever faster changing environment. This results in a need for enterprises to be able to innovate, and to adapt themselves quickly to these changes in the environment, and a desire to pro-actively exploit these developments in an attempt to create new business opportunities. This puts a major challenge on the enterprise's management to make the right decisions at the right time. To accommodate management in their decision-making and governance tasks, a new instrument is needed. This need is stressed even more by the complexities of the challenges and their consequences, as well as the diversity of stakeholders and their concerns.

The emerging instrument of *enterprise architecture* promises to provide management with insight and overview to harness complexity. Where classical approaches will handle problems one by one, *enterprise architecture* aims to deal with these issues in a coherent and integral fashion, while at the same time offering a medium to achieve a shared understanding and conceptualisation among all stakeholders involved and govern enterprise development based on this conceptualisation. As such, enterprise architecture plays a key role in the governance of organisations and their evolution. In this book, we therefore treat enterprise architecting as being an integral part of the governance of an enterprise's change and transformation processes.

Where the next Chapter aims to define the instrument of *enterprise architecture*, this Chapter aims to first provide an exploration of the motivations why enterprises turn to *enterprise architecture* as a means to find answers in their quest to meet modern day challenges. In doing so, we start with a brief exploration of the challenges (Section 2.2), which confront modern day enterprises. In Section 2.3, we then turn our attention to the stakeholders who have a stake in the enterprise and/or its development. By surveying the stakes and concerns of these stakeholders, we gain an understanding of the demands on enterprise architecture as a means of governing an

enterprise's change and transformation processes. We then continue, in Section 2.4, with a discussion of traditional approaches for strategy execution and governance. Section 2.5 then assesses to what extend these traditional approaches do indeed provide answers to the challenges of enterprises, leading to several shortcomings which are to be remedied by the use of *enterprise architecture*. Before concluding this Chapter (Section 2.7), Section 2.6 summarises the requirements on *enterprise architecture*.

2.2 Enterprises and their challenges

Enterprise are, as illustrated in Figure 2.1, confronted with a multitude of challenges. In this Section we discuss some of the challenges with which modern day enterprises are confronted, sparking the need for *enterprise architecture*. Discussing these examples also gives us an initial appreciation of the requirements that should be met by the means of *enterprise architecture*.

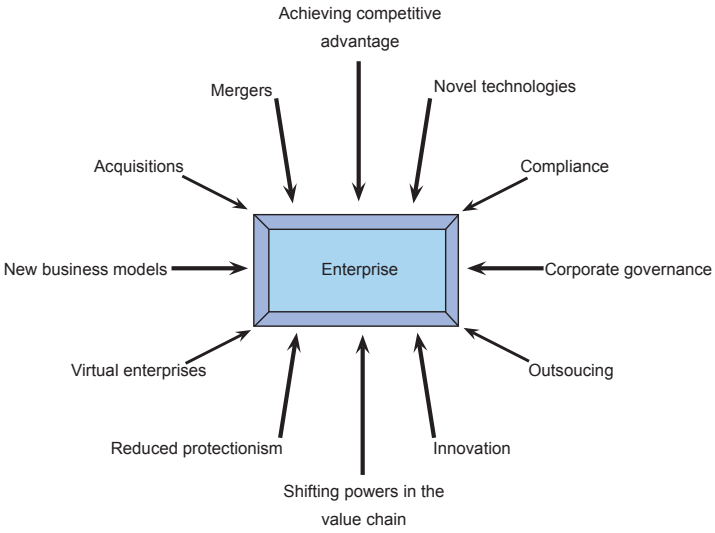

Fig. 2.1 Challenges confronting an enterprise

2.2.1 Keep up or perish

Enterprises face many changes, such as mergers, acquisitions, innovations, novel technologies, new business models, reduced protectionism, de-monopolisation of

markets, deregulation of international trade, privatisation of state owned companies, increased global competition, etcetera. These changes are fuelled even more by the advances of eCommerce, Networked Business, Virtual Enterprises, Mashups Corporations, the availability of resourcing on a global scale, etcetera [53, 58, 88, 134]. These factors all contribute towards an increasingly dynamic environment in which enterprises want to thrive. To improve their chances of survival, enterprises need to be agile. In other words, they need the ability to quickly adapt themselves to changes in their environment, and seize opportunities as they avail themselves. Such agility has become a business requirement in many lines of business, from the US army (schedules for combat systems from 8 years to 2 years) via the car industry (from thought to finish for a new model in a few months instead of 6 years) to the banking industry (time to market for a new product from 9-12 months to a few weeks [95]). Setting up new businesses has become a matter of hours, including online purchasing and payment systems. In practice, however, enterprises see themselves hampered in their ability to change in several ways:

- being uninformed about their own products, services, capabilities and internal structures,
- traditionally organisations were designed with efficiency and effectiveness in mind rather than agility,
- no common understanding and governance of key data resources,
- prevailing organisational structures, regulations, etcetera, have become engrained in the technological, social and cultural fabric of an enterprise,
- a plethora of legacy applications and infrastructures,
- duplicated functionality in terms of people and/or technology,
- interwoven and unclear responsibilities,
- organisational silos, self-contained business units who operate on their own, with no sharing of data,
- silo application, i.e. self-contained and isolated applications, which only provide functionality to a specific business process,
- old generation ERP systems embedded in the organisation's package based silos.

These impediments of change are usually the result of:

- wave after wave of mergers,
- results of projects intended as pilots evolve into structural parts of the organisation,
- new product introductions that have been conducted as insular projects,
- the swinging pendulum between centralisation and de-centralisation,
- (sub-)optimisation of development at a local level rather than at a more communal level, usually exacerbated by local profit and loss responsibilities.

Management needs insight in the ability of their organisations/enterprise to change as well as insight into ways of improving their agility by at least removing the impediments.

2.2.2 Shifting powers in the value chain

Clients of enterprises have become more demanding. A shift of power in the value chain is occurring. Clients have grown more powerful and demand customised, integrated and full life-cycle products and services. For example, rather than asking for a "forklift-insurance", they ask for "forklift-availability" in their warehouse. Instead of asking for a "printer", they require a guaranteed "printing service". Even more, customers have a tendency to ask for integrated service offerings. Rather than treating booking of a flight, a hotel, and a sight-seeing trip as separate services provided via separate outlets, customers opt for one-stop shopping. This is a shift from basic products to full services. The creation and delivery of such complex products and services requires additional competencies which may not be readily available within a single (pre-existing) enterprise. In this pursuit they increasingly engage in complex product-offerings involving other parties, leading to cross selling and co-branding, in which the request of the customer has a deeper impact in the complete value web. To ensure the quality of such products and services, a high level of integration and orchestration between the processes involved in delivering them is required. These developments trigger enterprises into re-organising themselves into specialized parts increasing the agility of the enterprise as a whole. Umar [141] introduces the notion of Next Generation Enterprises (NGE), which conduct business by utilizing innovative new business models. He claims such a NGE (known by names such as virtual enterprise, networked enterprise, real-time corporations etc.) will be the standard way of doing business, given its agility and ease of set up. Friedman [43] states that businesses are being formed not based on the core competencies they have, but instead on their ability to provide services by clever combinations of outsourcing and renting through service providers around the globe.

A relative new trend in NGE's is close collaboration on *research & development* as well as *innovation*. Enterprises that decide to structurally collaborate with partners in their innovation processes, or maybe even fully outsource innovation, are part of so-called *open innovation networks*. A leader in such networks is Philips. Philips carries out innovation projects in collaboration with partners within and outside their supply chain. Well-known examples are the Senseo coffee machine (partner: Sara Lee DEs) and the PerfectDraft (partner: Inbev). Management need insight into the opportunities and risks of collaborations, enabling informed decisions about creating new partnerships, joining or extending existing ones, or leaving one.

2.2.3 Comply or bust

In the networked economy, governance of enterprises becomes increasingly complex. One sees a shift in governance from individual departments within an organisation, to the entire organisation, and lately to the organisation's value web. Management does not only have to worry about the reputation of their own organisation, but also about the other organisations in their value web. How daunting the latter

might be can be illustrated by real life examples, such as a large shoe manufacturer who outsourced the production of shoes to another company, to only discover at a later stage that the latter made use of child labour. Although the latter company was not part of the shoe maker's own organisation, their reputation was still damaged, threatening their survival on the market-place.

Governance is not only an issue to an organisation on its own, but also a major concern to society as a whole. As a result of undesired and uncontrollable (side-)effects of the increased socio-economical complexity and interdependency of organisations, services, products and financial instruments. Recent examples of such side-effects are the well-known Enron scandal, as well as the sub-prime mortgage crises. To control and/or prevent such effects, new legislation has been put in place to better regulate enterprise practices. An example being the Sarbanes-Oxley Act [47] forcing enterprises to increase the quality of their governance and appropriateness of audits.

Management of organisations need insight into the compliance of their processes to their own goals as well as regulations provided by external regulators.

2.2.4 Achieving competitive advantage

Enterprises try to achieve and maintain a competitive advantage. In order to do so, they need to choose an optimal strategic position. Porter [102] distinguishes four basic units of competitive advantage: product development, purchasing, operation, and distribution of products or services. Performing these four activities better than one's rival is called *operational excellence*. Enterprises can, however, also opt for other ways of distinguishing themselves from their rivals. In [140], Treacy and Wiersema argue that enterprises should try and focus on one of the three disciplines of added value:

Product leadership – These enterprises aim to provide the best and/or most innovative products. An example would be Nike.

Operational excellence – These are typically enterprises, which strive to provide a basic level of service in the most efficient way. McDonalds would be a prototypical example of operational excellence.

Customer intimacy – Enterprises, which are customer focussed and aim to provide (complete) solutions for these customers. An example of such an enterprise would be Rolls Royce (the *car* manufacturer).

In the recent past, enterprises needed to excel in only one of the above areas to be successful, and meet industry standards on the other areas [140]. Due to the network economy and globalisation, there is a growing need to excel in a minimum of two areas (or at least in one and significantly improving in the other areas). To be able to make proper decisions in these crucial matters, management needs a clear view of the future and its impact on their enterprise.

2.2.5 Making technology the business differentiator

The evolution of information technology brings an abundance of new opportunities to enterprises. Technology becomes part of almost everything and most processes have become IT reliant, if not fully automated. Some recent illustrations of the innovative use of information technology to support pre-existing processes are:

- Delinquents which serve house arrest are monitored with RFID and GPS technology to make sure they do not leave their premises;
- Police officers in Groningen (The Netherlands) use a PDA during surveillance. This PDA, which is equipped with GPS and navigational functions, automatically alerts officers when they pass the address of someone with unpaid fines. The PDA also shows the positions of their colleagues [87].

The technological evolutions confront enterprises with the question of which technologies are relevant to the enterprise? Which technology should be replaced and which technology could be of use for developing new products (or services) or to enter new markets? Management needs insight in the features of new technology and the impact on, and possibilities for, their enterprise.

2.2.6 Excel or outsource

Increasingly enterprises outsource business processes. Outsourcing of business processes requires organisations to precisely understand and describe what needs to be outsourced, as well as the implementation of measures to ensure the quality of the outsourced processes [48, 94, 96, 115].

In deciding on what to outsource and how to safeguard its quality, management needs insight into the extent to which processes can be outsourced, the risks that may need to be managed when doing so, as well as the interdependencies within the outsourced processes and between the outsourced processes and the retained organisation.

Conversely, organisations with a strong tradition in a certain business process may decide to become industry leader for such processes. For example, processing of payments, management of IT infrastructure and logistics.

2.3 Stakeholders and their concerns

An enterprise has many stakeholders. Future development of an enterprise is likely to impact on the interests of these stakeholders. In this Section we briefly survey some classes of stakeholders and their specific concerns. In this book, we use the definition of stakeholder and concern as provided in [60]. A *stakeholder* is an individual, team, or organisation (or classes thereof) with interest in, or concerns relative

to, a system (such as an enterprise). *Concerns* are those interests, which pertain to the system's development, its operation or any other aspect that is critical or otherwise important to one or more stakeholders.

In making decisions about an enterprise's future directions, stakeholders want to obtain insight into the impact these directions will have on their concerns, and understand the risks involved in current and future initiatives. Even more, since present day enterprises are complex social systems of interrelated processes, people and technology, stakeholders are keen on finding a way to harness this complexity when judging the impact on their concerns.

Each type of stakeholder has its specific need for insight, control and overview. At the same time, they all want insight into the potential impact on the enterprise resulting from changes in its own strategy or its environment, and consequences of decisions about the enterprise's future directions. They also have the desire to communicate about these changes and impact. Communication will take place at enterprise level, business unit level, department level and project level depending on the responsibilities of the stakeholder involved in the communication. Below we briefly zoom in on the interests and concerns of three typical classes of stakeholders, and their needs, namely (see Figure 2.2):

1. stakeholders involved in a transformation;
2. stakeholders impacted by transformation results;
3. stakeholders sponsoring a transformation.

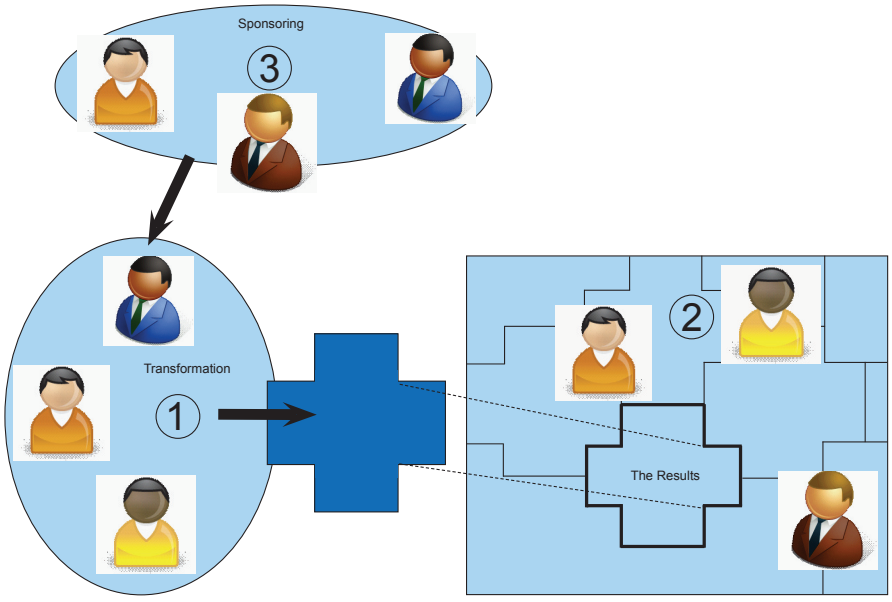

Fig. 2.2 Three stakeholder roles in a transformation

2.3.1 Stakeholders involved in a transformation

Stakeholders involved in the transformation of (parts of) the enterprise need insight in and control over the scope of their engagement. They need insight into such questions as:

- *What part of the enterprise will be impacted by the transformation?*
- *What are the boundaries of the part of the enterprise being transformed?*
- *What are the relations and dependencies with other transformations/projects?*

Furthermore, insight is needed into the contribution of their respective projects to the long-term strategy of the enterprise. This insight will prevent the implementation of solutions, which do not fit well into the overall (long-term) solution, even though they might be suitable in the context of a specific project. For example: the long-term strategy of some enterprise might be to apply an "enterprise service bus" for bilateral communication between applications enabling more flexible support of new business processes. If people involved in a specific project are not aware of this long-term strategy, it could be tempting for them to implement point-to-point solutions because this is probably less time consuming. Another example would be the long-term strategy to introduce shared service centres for services needed by different processes across the enterprise. If a project developing some new business process is not aware of this strategy, they may opt to realise some required sub-processes locally, which should have been provided by the shared service centre.

2.3.2 Stakeholders impacted by transformation results

A wide variety of stakeholders will possibly be impacted by the results of a transformation. Depending on the type of transformation, employees, operational managers, customers or business partners could be impacted. Typically they would be looking for answers to questions such as:

- *How will the new situation, resulting from the transformation, differ from my current way of working?*
- *How can I prepare myself for this new situation?*
- *What is the rationale for this transformation?*
- *When will the results of the transformation be effective?*
- *What type of change will happen and how to contribute to the realisation of the transformation?*

This insight gives these stakeholders understanding for the reasons and the effects of the transformation on their work, and knowing how to prepare themselves for the transformation results. As an example consider the transformation of an enterprise to become more customer oriented, leading to the introduction of a front-office where all customer contacts are handled, a back-office for handling customer requests, and a multi-channel service delivery. The stakeholders will clearly be impacted by this

transformation, and might want to know when they can fully use the Internet channel (customers), or what skills are required for the back-office workers (employees).

2.3.3 Stakeholders sponsoring a transformation

The discussion in the previous Section already stressed how decisions concerning the future direction of an enterprise (be it in terms of business aspects, human resources aspects or IT aspects) may have a profound impact on the future health of the enterprise as a whole. Therefore, it is of the utmost importance for the enterprise's management to have control over the decision-making processes as well as the desired transformations following from these decisions. Major trade-offs should be made explicit in terms of an evaluation of the alternatives. Management and other stakeholders are looking for some kind of "compass" or "atlas" that will guide them in making decisions about future directions of the enterprise, and will make clear to them (in their terminology) what the impact of future changes will be; at least to strategic management and stakeholders deemed relevant by the owners of the enterprise and/or sponsors of the major change or transformation process.

Fuelled by the challenges confronting enterprises, as discussed in the previous Section, some typical questions confronting (strategic) management are:

- *Are we able to deliver a new product? Which parts can be produced in house (by reusing current business services) and which parts should be outsourced (or produced by using external business services)?*
- *How sound is the business case a major transformation? What will it cost? How big are the benefits?*
- *What are consequences of alliances with external parties or innovation networks? What opportunities would such alliances offer?*
- *What is the impact of their decisions at different levels, such as enterprise level, business unit level and department level?*
- *What are the implications of major changes in the enterprise's environment, such as technology shifts, mergers or de-mergers, outsourcing or centralisation, and the introduction of new forms of legislation such as Sarbanes-Oxley act [47]?*
- *How does the current process landscape reflect our business priorities?*
- *To what extent might a specific project generate undesired side effects?*

2.3.4 Variety and complexity in dealing with stakeholders

As discussed in e.g. [78, 149], there is also an increased need to not only consider one aspect (such as business processes, IT, culture, human resources, knowledge domains, applications, etc) of an enterprise in isolation, but rather to see all aspects as being part of an integrated whole. The concerns of stakeholders, especially when considered in parallel, are hardly ever limited to one aspect only. Stakeholders will

want to gain insight into these aspects, their interdependencies and the possible impact of future developments on their concerns [78]. This means one has to deal with a variety of concerns when dealing with stakeholders.

In addition to the variety of concerns, one typically has to deal with a large number of stakeholders as well. This is commonly referred to by the notion of *social complexity* [31]. Social complexity is determined by:

1. the number of stakeholders involved,
2. the variety of their stakes and concerns,
3. the diversity of their functional, social and cultural backgrounds, and
4. the diversity of their communicative and cognitive abilities.

Effective communication can be very difficult when the social complexity is high. This calls for a shared meaning of key terms and concepts [19, 50, 144], as a prerequisite for a shared understanding of context, goals and issues and a shared commitment to the outcome.

2.4 Traditional approaches

In this Section, we consider two traditional approaches for dealing with the earlier discussed challenges. The first approach is the use of strategy to focus change and/or transformation efforts in an enterprise. The second approach is programmatic steering of change, involving governance, programme management, project management and portfolio management.

2.4.1 Strategy as a means to focus effort

In times of change, enterprises are hard pressed to make choices in order to survive. One of the disciplines that can be applied in making those choices at the enterprise level is strategic management. Strategic management is a combination of three main processes: strategy formulation, strategy implementation and strategy evaluation. It can be applied to coordinate the various aspects of management to enable an organisation to achieve its long term objectives.

Organisations are commonly defined as a goal-oriented cooperative of people and means [32]. There is no need to argue that organisations need some mechanism by which they can consciously make decisions about the way they deploy their resources. Even more, for organisations to sustain in the longer run, they need to make choices about their own future in relation to the environment. Organisations typically use *strategies* to focus their resources and efforts towards the achievement of goals. Based on reflections on future evolutions, those strategies express choices for main directions of their organisation.

The concept of strategy is usually related to other concepts, such as: mission, vision, goal and policies. Several definitions of these concepts are in existence, for example:

Mission – Overriding purpose in line with the values or expectations of stakeholders;
Vision – Desired future state: the aspiration of the organisation;
Strategy – Long-term direction;
Goal – General statement of aim or purpose;
Policy – A statement giving direction towards the achievement of goals.

In [68], a strategy is positioned as the resultant of an organisation's mission and vision, while goals are formulated as concrete milestones towards the achievement of an organisation's vision. These terms can be related to each other, and positioned in three tiers (see Figure 2.3):

1. On the first tier we find **mission**, the reason of existence for an organisation;
2. The second tier provides a concretisation of the mission in terms of a **vision** and a **strategy** aiming to realize the vision;
3. The third tier provides a further concretisation by refining the vision to **goals** and the strategy to **policies**.

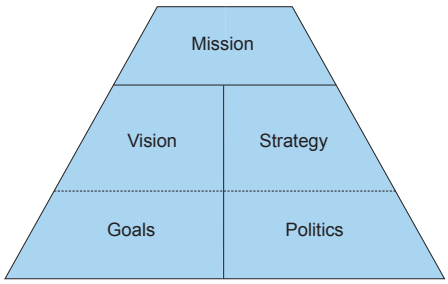

Fig. 2.3 Three tiers of strategy formulation

The distinction between vision and goals and analogous between strategy and policies is not absolute but gradual. In Figure 2.3 this is illustrated by the dashed bordering.

To further illustrate these concepts, consider the following example. A mortgage company has defined as its mission "*to supply superior financial solutions to make individuals live in a better place*". As a vision it aspires to be "*the largest and internationally recognized mortgage banking firm*". As part of their strategy they have decided that their way of becoming the largest mortgage firm is not by means of organic growth, but rather by means of take-overs of other mortgage firms. The policy they have set out for the take-overs is: "*only friendly take-overs in countries in which the organisation does not yet operate*". The goal in which they intend to

measure the achievement of their mission is to grow their market-share by 2% each year.

The execution of a strategy is a continuous process, in particular because the strategy itself is likely to evolve continuously as well. It is necessary to permanently improve and adapt an organisation's strategy due to changes in either the environment of the organisation or changes in the organisation itself. This leads us to discern two flavours of strategy evolution:

Outside-in – A strategy can be influenced by developments in the environment of an organisation [68]. These developments may be due to political, environmental, socio-cultural, technical, economic, as well as legal factors (also refer to the PESTEL framework [68]). Some examples of developments and their possible influence on strategy are:

1. Following on the open electricity market in Europe, electricity companies in the Netherlands are required to unbundle the ownership of network facilities and the delivery of electricity.
2. Technology innovations such as RFID and GPS/Galileo have enabled new applications and even new forms of doing business, as well as product innovations such as unmanned harvesting in an agricultural context and innovations in the car industry.

Inside-out – A strategy can also be influenced by changes in the availability (and becoming aware) of an organisation's resources and competences [68], leading to a resource-based view of a strategy. Some example of developments in this area and its strategic implications are:

1. Having specialists in a specific field and positioning that field as a key business driver. Consider, for example, Porsche. Porsche is a company, which traditionally focuses on the design and construction of fast cars, is now also responsible for the design of several other artefacts such as domestic appliances, thus transplanting their design quality and image to other products.
2. Organising a number of services in a shared service centre in order to improve quality and lower costs of these services [10, 62].

When related to the traditional SWOT analysis, summarising key issues that are most likely to impact on an organisation's strategy development, the outside-in approach covers the Opportunities and Threats, while the inside-out approach deals with Strengths and Weaknesses.

2.4.2 Programmatic steering of change

Typically the implementation of a strategy is executed through programmes. A programme can be considered as a layer above individual projects. It consists of multiple inter-dependent projects that together deliver some defined objective(s) for an organisation. The objectives of a programme are typically at a strategic level, aiming

to achieve benefits and improvements in the business. We now will briefly explain what instruments are available for programmatic steering of change, namely governance, project management, portfolio management, and programme management.

Governance – as discussed in Subsection 2.2.3, is a business challenge, which becomes increasingly important in the networked economy. To make over-all change and governance processes of the enterprise possible, governance needs a clear insight into the substance and coherence of the entire value chain.

Projects – aim to realize parts of the *to-be* situation. In doing so, the projects need to have a clear view of the *to-be* and the *as-is* situation. In the end, project results should not only answer the concerns of the project's stakeholders, but it should also be aligned to strategic directions and constraints.

Portfolio management – is a means to manage initiatives and programmes in an integrated, coherent fashion. For defining these programmes, a common language is needed for the business and IT aspects involved and also for (expected and realized) outcomes and added value.

Programme management – is an instrument to achieve business benefits. It is a way to manage uncertainties, changes and coherence between projects. In coherence with the complete portfolio, rankings for the projects or interventions have to be made to develop a route consisting of projects towards the desired state. Programme management focuses on the managerial aspects of this body of projects. Take for example the Goal-Efforts-Network [133] method for programme management. This method prescribes an approach to take in deriving programme activities, bundled in projects, from the goals of a transformation. This will ensure the goal-orientation of those projects, but not necessarily the coherence in work done by these projects; indeed, several projects may want to change the same things – even in different directions – in order to reach different goals. Therefore programme management needs insight in the cohesion between the product aspects of these projects.

2.5 Assessing traditional approaches

In the previous Section, we introduced strategic management and programmatic steering of change as answers on the challenges confronting enterprises. In this Section we concern ourselves with the question: *To what extent do these traditional approaches answer the challenges indeed? Are these approaches sufficient?*

2.5.1 Putting strategy into action

A strategy is essentially a high-level choice of the way an organisation aims to achieve their mission and vision. This immediately raises the question: *How can this strategy be executed?* Given the definitions above, the obvious thing to do is to

refine the vision and strategy to more specific goals and policies. This is, however, not enough since this does not yet provide an operational perspective on *how* to indeed achieve these goals. Therefore, in addition to the identification of goals and policies, a number of programmes, projects or activities are needed to indeed implement the strategy. This Section examines this process of strategy execution and implementation, and shows the major issues that could arise. We take the stance that a *proper definition of a strategy* should be specific, unambiguous, achievable, relevant and actionable, and be based on a profound insight into the impact of change. For such a proper definition of a strategy, and the execution of such a strategy, an additional means is needed: enterprise architecture.

Translating a strategy formulation into strategy execution or, as it is also called, *strategy into action*, is concerned with three major areas: organising the organisation, resources allocation, and change management [68]. "*Organising the organisation*" takes care of structuring an organisation's structure, processes, and relationships to support successful performance. "*Resources allocation*" is the enablement of success by how various business areas support strategies and vice-versa. "*Change management*" addresses the individual and organisational issues concerned in managing change.

Within each of these three areas problems pop up during strategy execution. Not much scientific literature exists uncovering the causes of such problems. Therefore, we have gathered a list of potential causes derived from our best practices. Besides that, we found one source (1), which concentrates on such causes, as well as four others (2-5) who focus on ways to improve strategies:

1. the Free University of Amsterdam and consultancy agency Turner, who show seven ways to "screw up a strategy" [125];
2. Kaplan et al. [71] who introduce the balanced scorecard to map strategy;
3. Zagotta et al. [156], who give seven keys to successful strategy execution;
4. a study into the real stories behind mergers and acquisitions [3];
5. and a McKinsey research about what drives successful transformation in organisational performance [103].

We combined these sources with the causes we already found ourselves and mapped these in the three areas "*organising the organisation*", "*resources allocation*", and "*change management*". The result of this mapping is described below. The causes identified by us have been represented in *italics*.

With regards to "organising the organisation", one will come across causes such as *a vague vision and / or strategy*and *alternatives that are not shown or balanced*. These causes are recognized by [71] who argue that many top executives only give very limited information to their employees. In an attempt to counter these causes, [156] introduce two strategies represented by two key phrases: "quantify the vision" and "plan what you are not going to do". The first phrase represents the will to transform corporate hopes and dreams into tangible targets, while the second represents the need to show what initiatives should *not* be executed because of the new strategy. The first two of the causes raised in [125], being "discord at the top" and "let's just start" address the disarray that may result when a vision or strategy

is not clear enough. “Discord at the top” really signifies that among management there is already ample room for multiple interpretations of the directions set out in the strategy, while not all managers propagate the same goals underlying the vision. The “let’s just start” phrase captures the fact that quite often it is forgotten to also identify what you want to reach and how. Based on our own experiences, we would like to add some additional concerns:

- *decision-making that is done too early or too late*, which often results in wrong decisions;
- *a strategy without freedom of choice* which limits the execution teams too much;
- *a vision and strategy that are not well-defined, causing different interpretations and at execution level;*
- and *having solutions that cannot be traced back to the strategy*, which makes it difficult to show the added value of the project and the results of the strategy.

With regards to “resources allocation”, one comes across problems such as *solutions that do not fit in* because departments may have a tendency to make their own plans for those parts of corporate strategy that are relevant to them, while not integrating these plans with other departments. Alternatively, people within might work on the realisation of one aspect of the strategy, while not being aware of the relationships (cohesion!) to other aspects. “Resources allocation” also refers to possible shortages-of / struggles-over resources. For example, *when priorities are not clearly set, a fight for money can occur*, or *during strategy execution, it is realised that the strategy is not feasible or realistic in resources*. Two of the ways to “screw up a strategy” in [125] concentrate on the latter causes, specifically “flexible in execution” and “leave it to the stakeholders”. The first potential stumbling block deals with the way the strategy is to be executed, while the second addresses the requirement that the parties involved in the strategy execution need to know what is expected from them. In support of this, [3] mentions “clarifying organisational structures and governance processes” as one of the most important issues to address during mergers.

“Change management” is concerned with the management of change processes taking place in an organisation. What may happen during the change process is that *the strategy disappears in a closet for a couple of years*. In other words, the strategy itself is not changed during the change processes, as if the enterprise’s socio-economic and technical environment is waiting patiently for the enterprise to catch up. The authors of [156] recognize this and aspire an open strategy approach, which entails that employees need to work *with* the strategy in terms of an ongoing process of reviewing and maintaining strategic progress. Johnson et al. [68] also identify that change programmes need to be active and vivid within organisations, otherwise it is a risk that the employees are going to see such changes as rituals signifying very little. In [125] it is stated that it does not suffice to appoint a change manager, but rather that a change *leader* is called for, someone who can make a difference. Two other typical strategy killers mentioned in this research are “force change upon the organisation” and “send everyone to courses”. These are typical examples of “uncontrolled and uncoordinated efforts” [68] which will not be understood by the people in the

organisation. Top management also plays a crucial role in organisational changes. On the one hand, an organisation needs *management commitment* to successfully execute a strategy [3] and on the other hand, the change should not be invented by senior executives only. The latter would lead to "ivory tower change" [68]. *A lack of communication* and *a lack of supervision* are two causes that also relate to change management. Change concerns the whole organisation and the whole organisation needs to be involved in this change. Therefore, communication and visibility are very important. [103] held a survey in which these kinds of change management mechanisms were questioned. [156] prescribe mantras for communicating strategy; simple lines to communicate the essence. The last pitfalls we will mention are:

- *Endless strategy formulation*, essentially a relative to *analysis paralysis*. This pitfall will usually lead to a situation in which the execution of the strategy is not attained.
- *Under the name of strategy many different projects arise* that actually have no benefit to the strategy. Johnson et al. [68] refer to this as "hijacked processes".

In all three areas (organising the organisation, resources allocation, and change management) causes pop up which need to be addressed. Moving from strategy formulation to strategy execution is not a simple path to follow. Many enterprises struggle with their strategy execution and need a means to support this.

2.5.2 Putting programmatic steering into action

Programme management caters for change management, effectiveness and the control of time and budget. By handling problems one by one, such solution development becomes phased and manageable. At the same time best practices on programmatic steering show the following common shortcomings and needs:

- portfolio, programmes and projects don't stay in line with agreed strategy and constraints;
- programme / project sequence planning is not solidly or explicitly underpinned:
  - e.g. the programmes in year 2 finds out that part of the solutions of the programmes in year 1 is superfluous or could have been simplified;
  - e.g. it is not always known that the result of project 1 is required for execution of the solution to be delivered by project 2;
- realised solutions overlap or are incomplete;
- realised solutions are incompatible with each other, with solutions in the context or with acknowledged business and IT policies;
- realised solutions are optimal for their project, but not the best for the enterprise as a whole;
- programmes in the change portfolio interfere, because of lack of common language, e.g. on *to-be* and *as-is* situation or on added value and outcomes;
- the business case for an intended transformation is not complete;

- business attention for programmes directly focused at implementing a business initiative is ensured, neglecting programmes to ensure the required boundary constraints;
- the quality of the end result is traded off with duration and budget, thus unconsciously downgrading the result to a mid-term instead of a long-term solution;
- the same requirements are differently elaborated and solved by different projects.

Some examples of consequences of these shortcomings are:

- the enterprise is not fully obtaining the business benefits that the programme was set out to achieve;
- surprises in systems management costs;
- lack of interoperability and consistency;
- lack of economies of scale (through common use);
- lack of a overall consistent experience of the systems.

2.6 Requirements on enterprise architecture

The road from strategy formulation to strategy execution, including the use of programmatic steering, is certainly not an easy one to travel. Research shows that less than 60% of the strategic objectives in organisations are reached [125]. When considering the possible failures in strategy execution as discussed in the previous Section, an instrument is needed to support this process. As illustrated in Figure 2.4, this instrument is positioned between strategy and programme management. This Section describes the requirements to this additional instrument. We start with a reflection of the requirements following from the causes for problem areas in strategy execution: "*organising the organisation*", "*resources allocation*", and "*change management*".

The causes involving "organising the organisation" call for a means that makes the strategy more specific, unambiguous, achievable, relevant and actionable, while at the same time providing an overview of the desired future state and the impact of change with respect to the current state. As [19] show, it is important to keep an eye out for ambiguities during development projects, i.e. the need to define terms precisely. For example, what does a given government agency mean by *customer-oriented*? Is that to be interpreted as serving the citizen, to service the responsible minister, or both? After defining such terminology, the programmes and projects needed to arrive at the desired state need to be "designed". To be able to do so, different alternatives to obtain the state have to be elaborated, evaluated, and decided upon. A method to structure and document these alternatives is necessary, just as tracing of the contributions of individual proposed projects to the realisation of the strategy. Furthermore, all stakeholders are likely to want insight into the key issues to be able to make decisions. To obtain this insight, views are needed which highlight the important issues to the specific stakeholders. The added value of each of

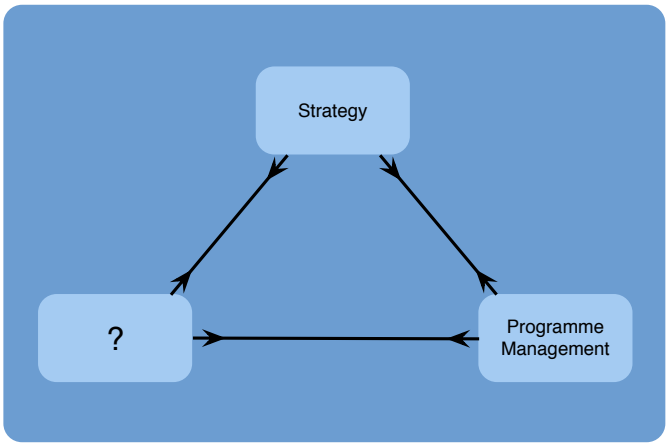

Fig. 2.4 The gap to be filled by enterprise architecture

the proposed programmes and projects needs to be assessed from an agreed and committed to perspective [78, 108].

In the second area of "resources allocation" it is essential to look at the different aspects of the business areas in cohesion. Therefore, concrete plans of the desired state and the way towards it are required and a prioritisation in the programmes and projects based on time, resources, and goals is needed. Cohesion is needed in the business processes and serving of the client (e.g. don't have information on traffic conditions end at the border of a region, but let it be in the perspective of the driver who travels from A to B), in the information required (e.g. each actor has all information available required for his function), and in the infrastructure (the infrastructure is connected in a meaningful way for end users).

The "change management" area strives for a common understanding and shared commitment between all stakeholders involved. This requires stepwise commitment together with growing insight in impact of change. Therefore, effective communication is needed towards all stakeholders. A communication breakdown is required in which a common language and models are used to communicate between all parties [19, 107]. In this area it is crucial to address the culture of the organisation and take care of the commitment and recognition of the stakeholders. It is necessary to deal with conflicting requirements and to adapt quickly to changes in the situation.

The route to be taken from strategy formulation to execution, including the use of programmatic steering, needs a means that enables it to do the right things (be effective) and to do things right (be efficient) in strategy execution. Therefore, the means needs to be a tool for steering, coordination as well as communication. Using this means, it should be possible to:

1. Gain insight into the current state of the enterprise at a suitable abstraction level to understand and to analyse issues that hamper the execution of the strategy of the enterprise;

2. Gain insight into the current state of the enterprise to assess its compliance to (external) regulations;
3. Deal with social complexity of stakeholders involved in enterprise transformations;
4. Develop a business case for the chosen strategic direction;
5. Explore strategic alternatives for the future direction of the enterprise, while considering issues, challenges, feasibility and impacts, and eventually making a decision for an alternative of choice;
6. Express/depict a coherent, comprehensive and concrete image of the desired future state(s) of the enterprise;
7. Design a roadmap for the transformation;
8. Distinguish between short-term solutions and long-term (structural) solutions;
9. Give a clear context and direction – limiting design freedom – of individual projects that contribute to the transformation;
10. Select available solutions and/or packages that are to remain or to become a part of the solution, whether in-house or sourced by a business partner;
11. Guard the proper execution of any transformation project to be in line with the strategic direction (or to be knowingly informed that it deviates) and with external regulations;
12. Provide a common language to a portfolio of changes/transformations of an enterprise;
13. Enable traceability of design decisions from the strategic level via programs to specific projects.

In addition to the above requirements on enterprise architecture as a means, based on the discussions in this Chapter, we also identify seven key applications for enterprise architecture:

1. Investigate problems/shortcomings in a pre-existing situation, including the creation of a shared (among stakeholders) understanding of the existing situation;
2. Express (and motivate) the future direction of an enterprise, as well as investigate (and evaluate) different alternatives. This also involves the creation of a shared (among stakeholders) conceptualisation of the (possible) future directions, and shared agreement for the selected alternative;
3. Identify key problems, challenges, issues, impediments, chances, etcetera, as well as make well motivated design decisions that enable a move from the existing situation into the desired strategic direction;
4. Provide boundaries and identify plateaus (intermediary steps) for the transformation of the enterprise towards the articulated strategic direction. In this context, enterprise architecture is used as a planning tool, making the realisation of a strategy more tangible;
5. Give a clear context and direction for a portfolio of projects working towards the realisation of the first plateau as defined at the tactical planning level;
6. Select one or more standard solutions and/or packages that are to become part of the solution and/or decide to outsource an entire business process/service to another enterprise;

7. Create the high level design of an actual step in the enterprise transformation as it will be realized (and implemented) in the context of a specific project.

2.7 Summary

In this Chapter we have explored the motivations why enterprises turn to *enterprise architecture* as a means to find answers in their quest to meet modern day challenges such as the constantly evolving environment in which they need to operate, outsourcing, network organisations, etc. Given these challenges, we then turned our attention to the stakeholders and examined their stakes and concerns, their needs with regard to an enterprise transformation. Management, for example, needs insight into the impact of changes, alternatives, technological developments, new government regulations, etc. Stakeholders involved in enterprise transformations need, for instance, insight into the boundaries of systems/processes to be developed and the relation to adjacent systems/processes. Stakeholders involved in the outcome of a transformation typically want to gain insight into the impact of the new situation on their work and personal goals. We also discussed the notion of social complexity as a function of the number of stakeholders involved, variety of concerns, and the diversity in their backgrounds and abilities.

Before claiming a place for enterprise architecture as a new instrument fitting a need, we discussed some traditional approaches as well as their shortcomings for *putting strategy to action* and *programmatic steering*. We mapped the causes for these shortcomings to the areas of "*organising the organisation*", "*resources allocation*", and "*change management*". This discussion, finally led to the identification of high-level requirements on the new instrument of *enterprise architecture.*

2.8 Discussion statements

1. The introduction of enterprise architecture heralds the end of strategy.
2. Enterprise architecture should be in the lead for portfolio and programme management.
3. No vision, no architecture.
4. No architecture, no vision.
5. Enterprise architecture is only necessary for major changes in strategy that effect the entire organisation.
6. Enterprise architecture is the only constant in ever changing enterprises.

Chapter 3
Positioning Enterprise Architecture

In the previous Chapter, we have discussed the needs for enterprise architecture. This Chapter is concerned with enterprise architecture as a means to meet these needs. We will start this Chapter with a historical perspective on the concept of architecture as a means of obtaining insight into, as well as harnessing, complexity. To gain a better insight into the role of enterprise architecture in governing transformations, Section 3.2 will then discuss the *governance paradigm* and relate this to the role of enterprise architecture. Based on this discussion, Section 3.3 then continues by identifying seven possible applications of enterprise architecture from a governance perspective. Using this as a context, Section 3.4 provides a discussion of several definitions of enterprise architecture, while also providing the definition of *enterprise architecture* as used in this book. To make this definition more specific and tangible, Section 3.5 will discuss the key concepts underlying enterprise architecture, while Section 3.6 will highlight the benefits of enterprise architecture in relation to the needs identified in the previous Chapter. Finally, Section 3.7 takes a first brief look at the competencies needed from the architect.

3.1 A historical perspective on enterprise architecture

The recorded history of classical architecting began more than 4000 years ago in Egypt with the erection of the pyramids, the complexity of which had been overwhelming designers and builders alike [82, 113]. This complexity had at its roots in the phenomenon that as systems became increasingly more ambitious, the number of interrelationships among the elements increased far faster than the number of elements themselves. Pyramids were no longer simple burial sites; they had to be demonstrations of political and religious power, secure repositories of god-like rulers and their wealth, and impressive engineering accomplishments. Each of demands, in itself, already required major resources. The complex interrelationships among combined elements were well beyond what the traditional tools of the engineers and builders could handle. This led to the introduction of architecture as

a means to obtain and maintain insight into these complex relationships. We have remained doing so. Following the evolution of our societies, we have used architecture as a means of obtaining insight and harnessing complexity of a wide variety of constructs as illustrated in Figure 3.1.

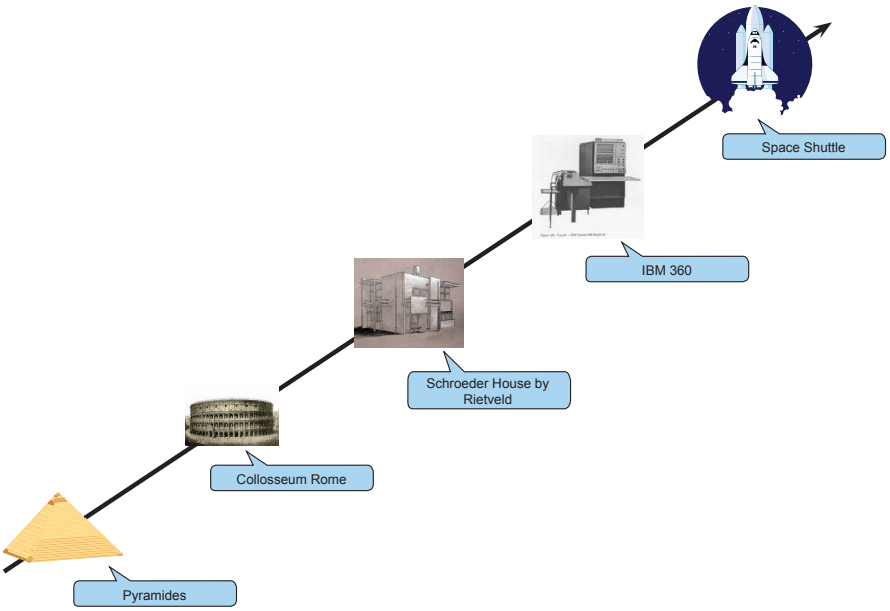

Fig. 3.1 Evolution of constructions

After years of architecture in the physical world, the term has also taken a foothold in the field of IT. Architecture is well known from the world of construction. Therefore some twenty years ago, the IT industry became confronted with complex structures and decision, a comparison with the construction industry seemed an obvious one. Probably the first person to use the term architecture in this context was Gerrit Blaauw [6]:"*The term architecture is used here to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behaviour, as distinct from the organisation of the data flow and controls, the logical design, and the physical implementation.*" Gerrit Blaauw was the co-developer of the IBM 360 computer family in the nineteen sixties. In his publications he refers to the architecture (i.e. design) of computers, while discussing such topics as modularity, reliability and consistency. At about the same time, Edsger Dijkstra started his work on structured programming. Although he did not use the word architecture, he repeatedly underlined the importance of the structure of software, thus laying certain foundations for architecture. This comparison leads to terms such as software engineering and structured programming. At that time, this

comparison brought a degree of order into many aspects of the creation of these programs as advocated by David Parnas [100].

When software applications became larger and larger, people such as Mary Shaw and David Garlan coined the term software architecture [123]. This notion of architecture deals with the key design principles underlying software artefacts. In the 1980's and 1990's people became aware that the development of IT (information technology) should be done in conjunction with the development of the context in which it was used. This led to the identification of the so-called Business/IT alignment problem [55, 99, 135]. Solving the Business/IT alignment problem requires enterprises to align human, organisational, informational and technological aspects of systems. Quite early on, the term architecture was also introduced as a means to further alignment, and thus analyses and solves Business/IT alignment problems [21, 135, 155]. Recently the awareness emerged that alignment between business and IT is not enough; there are many more aspects in the enterprise in need of alignment. This has led to the use of the term architecture at the enterprise level: Enterprise Architecture [18, 20, 37, 78].

3.2 Governance paradigm

According to [1], governance is "*the activity of [] controlling a company or an organisation*" or in other words the supervision of the compliance of rules. In our view, enterprise architecting is an integral part of the governance of an enterprise and its transformation.

Ideally, an enterprise architecture plays a pivotal role in the continuous improvement process of an enterprise. In order to better understand the governing role of enterprise architecture, this Section offers a discussion of the governance paradigm [79] and consequently applies it to an enterprise transformation context. Figure 3.2, which is based on [79], depicts the basic governance paradigm. The governance paradigm involves three important assumptions:

1. there is some system[1], the target system, which interacts with its environment;
2. this target system needs to be governed;
3. there is another system, the governing system which does the actual governing.

The essence of the governance paradigm is that during the realisation of a process there is some kind of interaction with the environment (input and output), and that this process is controlled by some (internal) authority which monitors, and if necessary adjusts, the process to make sure the intended objectives are reached. This authority is called governing system[2] (GS). The system governed by the GS is re-

[1] Note: system here is to be understood in its original sense of the term [11], and not as a synonym to application system as is the case in software development. In the context of enterprise architecture, we are specifically interested in active systems [28].

[2] Note that the original governance paradigm used Dutch terminology. In [80], an English translation can be found using the term target system and controlling organ. Since in the field of enterprise

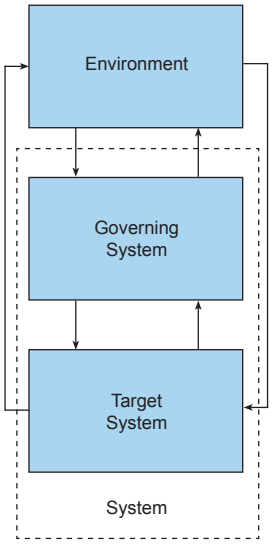

Fig. 3.2 The basic governance paradigm

ferred to as the target system (TS). Since an organisation is part of a larger system, the GS also interacts with the environment to determine which services of products to deliver, to determine new opportunities and to determine changes in the environment.

In the case of enterprise architecting the target system that needs governing is the transformation process of the enterprise, where not-transforming, i.e. maintaining a status quo is considered as a special transformation process. The latter case may actually take more effort than one would expect. Maintaining a status quo requires activities preventing erosion of an existing structure. Taking the enterprise as the target system, leads to the situation as depicted in Figure 3.3. In an operational enterprise, a distinction is made between a target system comprising the operational processes and a governing system, which governs these operational processes. The operational enterprise is transformed (to better meet the challenges and opportunities posed by its environment) by an enterprise transformation system. This transformation system is comprised of a transformation governing system and the actual transformation process(es). These latter processes constitute the target of the transformation governing system.

As mentioned before, enterprise architecting should be regarded as being a part of the governance of the enterprise transformation. Figure 3.4, therefore, shows a refined view on the governance of an enterprise's transformation processes involving three sub-domains: strategy, architecture and programme management [116].

---

architecture the term governance is used rather than controlling, we prefer to use term governing rather than controlling. To also stress the fact that the governing organ really is a system, we shall actually use the term *governing system*.

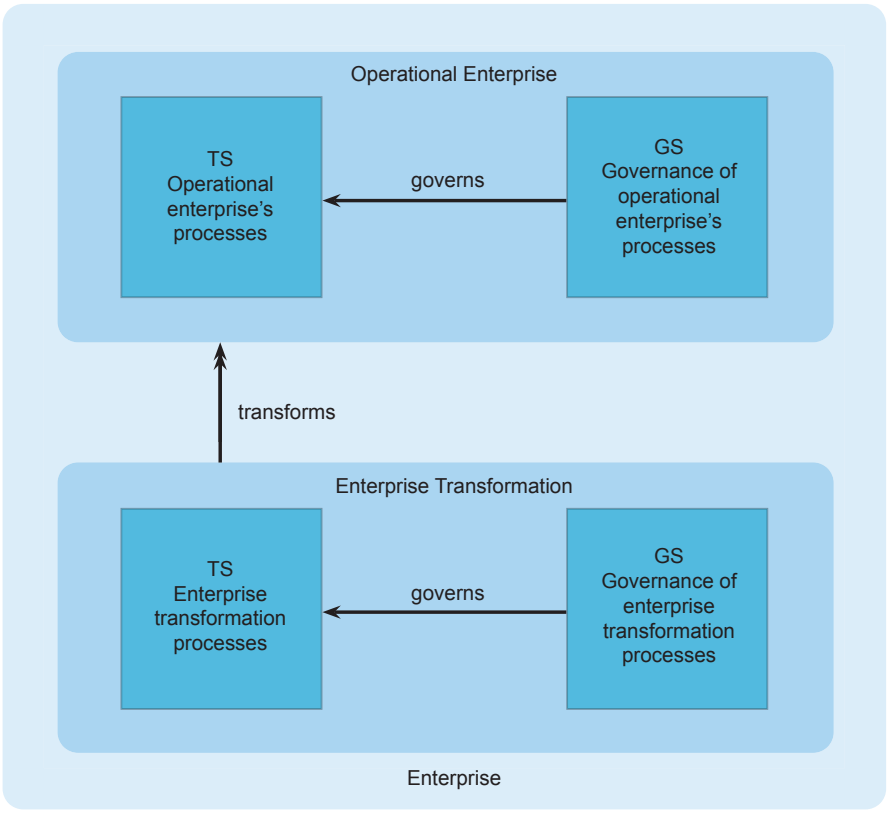

Fig. 3.3 Governance of an enterprise's transformation

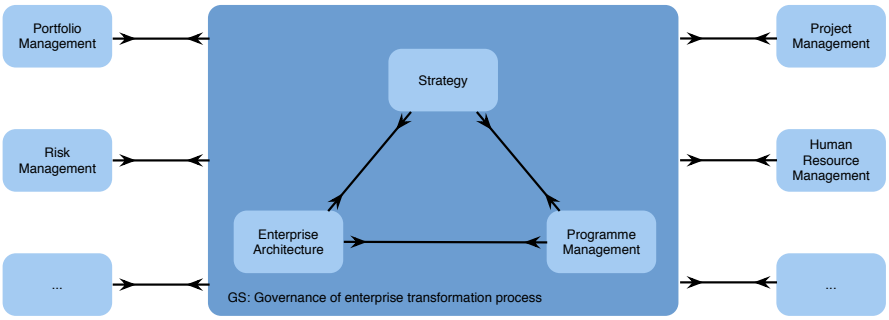

Fig. 3.4 The role of enterprise architecture

Based on the requirements on enterprise architecture as a means, as discussed in the previous Chapter, enterprise architecting can be likened to the use of a "dashboard" which allows the architect and stakeholders to steer the enterprise's transformation processes. When using the dashboard as a metaphor, the "*dashboard*" displays the *enterprise architecture* in terms of relevant aspects of the current state of the enterprise, its future direction and the desired states of the enterprise. Just as the selected/displayed speed, altitude and direction of an airplane is not *the dashboard*, but rather *displayed on* the dashboard, the dashboard is not the enterprise architecture. Analogously, it is the enterprise architecture, or rather a part thereof, what will be displayed on the dashboard. In addition, the dashboard may contain a report on the gaps between the current state and desired states, as well as its *operational performance* in terms of its current state.

In an airplane, a "*dashboard*" comprises of indicators (meters, lights, etcetera) and controls (levers, handles, pedals, and knobs). In the case of enterprise architecture as a means to govern transformations, the dashboard needs at least:

- *indicators* giving insight into:
  - the enterprise's current state,
  - the enterprise's future state,
  - the enterprise's current performance,
  - the enterprise's future (expected) performance,
  - the direction and progress of its transformation processes,
- *controls* allowing the transformation processes to be influenced.

The indicators may take the form of models, views, performance measurements, etcetera. The controls may take the form of (enforced) reference models, design principles, standards, etcetera. This is illustrated in Figure 3.5. The process of measuring, providing insight, decision-making and directing the enterprise's transformation process is a continuous (and far from linear) process. Based on the insights provided by from the dashboard, the stakeholders in conjunction with the architect may decide to adjust the directions as set out on the dashboard.

The situation depicted in Figure 3.5 is still somewhat naive in the sense that it takes a rather reactive perspective. If the architect and stakeholders would have some kind of a predictive model, which predicts future properties of the enterprise, the transformation processes, and their environment (eco-system), then this model can be used to more proactively steer the transformation process of the enterprise. This leads to the situation as shown in Figure 3.6. Using a model of possible target systems, the enterprise system and the ecosystem in which they operate, what-if analysis can be conducted based upon which the actual transformation processes can be directed more pro-actively. This essentially leads to an experimentation environment with a shadow dashboard and shadow (eco)system. This experimentation environment will provide the stakeholder with insight in the impact of change, based on different scenarios.

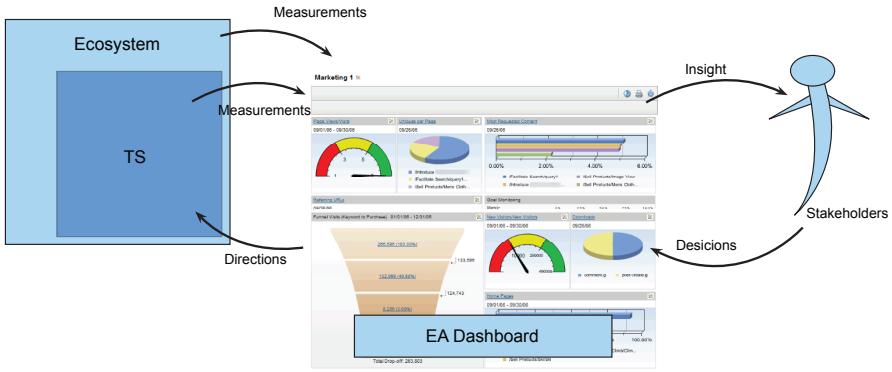

Fig. 3.5 Enterprise architecture on a dashboard

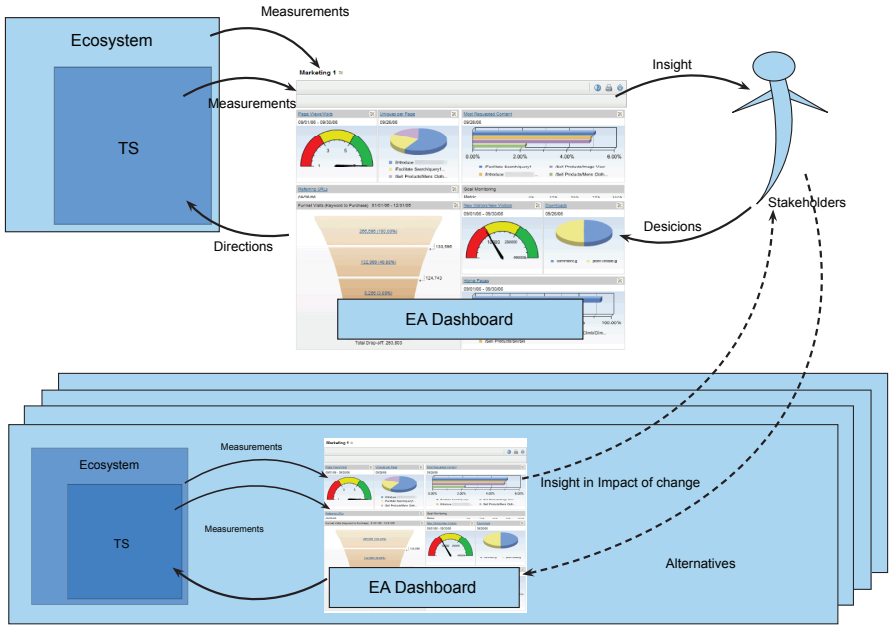

Fig. 3.6 Pro-active governing of transformation process

3.3 Key applications for enterprise architecture

Based on the needs and challenges of enterprises as discussed in the previous Chapter (in particular Section 2.6) we identify seven key applications for enterprise architecture as a means. In combination, these applications provide an instrument to make informed decisions as well as to ensure compliance of the transformation to these decisions, at several levels of specificity:

Situation description – Use enterprise architecture as a means for goal/cause analysis to investigate problems/shortcomings in an existing situation. This also involves the creation of a shared (among stakeholders) understanding of the existing situation.

Strategic direction – Use enterprise architecture to express (and motivate) the future direction of an enterprise, as well as investigate (and evaluate) different alternatives. This also involves the creation of a shared (among stakeholders) conceptualisation of the (possible) future directions, and shared agreement for the selected alternative.

Gap analysis – Use enterprise architecture to identify key problems, challenges, issues, impediments, chances, threats, etcetera, as well as make well motivated design decisions that enable a move from the existing situation into the desired strategic direction.

Tactical planning – Use enterprise architecture to provide boundaries and identify plateaus (intermediary steps) for the transformation of the enterprise towards the articulated strategic direction. In this context, enterprise architecture is used as a planning tool, making the realisation of a strategy more tangible.

Operational planning – Use enterprise architecture to give a clear context and direction for a portfolio of projects working towards the realisation of the first plateau as defined at the tactical planning level.

Selection of partial solutions – Use enterprise architecture as a means to select one or more standard solutions and/or packages that are to become part of the solution and/or decide to outsource an entire business process/service to another enterprise.

Solution architecture – Use enterprise architecture to create the high level design of an actual step in the enterprise transformation as it will be realised (and implemented) in the context of a specific project.

In Figure 3.7, we have illustrated these seven application areas. Each of these seven application areas will yield different enterprise architectures, which are clearly interdependent. By ensuring compliance among these architectures, governance and informed decision-making, from the strategic level to the operational level is enabled.

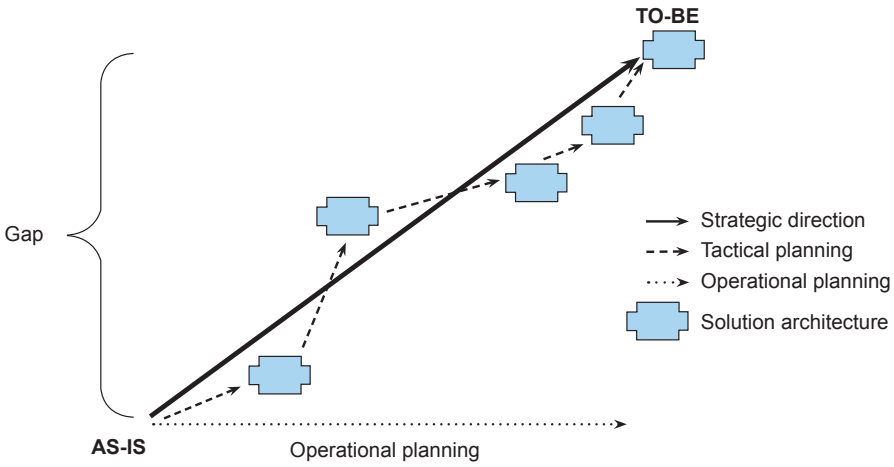

Fig. 3.7 Applications for enterprise architecture

3.4 Defining enterprise architecture

The previous Sections will undoubtedly already have shed some light on what we regard as enterprise architecture. In this Section we will make this more specific by providing our own definition of this concept.

3.4.1 Definitions of enterprise architecture

Before providing our definition of enterprise architecture we start with a discussion of some of the existing definitions of IT/information/enterprise architecture:

- The *Institute of Electrical and Electronics Engineers (IEEE)* defines architecture as: "*An architecture is the fundamental organisation of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution*" [60].
- The *Open Group's Architectural Framework (TOGAF)* defines architecture as: "*Architecture has two meanings depending upon its contextual usage: (1) A formal description of a system, or a detailed plan of the system at component level to guide its implementation; (2) The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time*" [139].
- The *Clinger-Cohen Act's* definition of IT Architecture is: "*The term 'information technology architecture', with respect to an executive agency, means an integrated framework for evolving or maintaining existing information technol-*

ogy and acquiring new information technology to achieve the agency's strategic goals and information resources management goals" [142].

- The *Netherlands Architecture Forum (NAF)*, defines architecture conceptually as "*a normative restriction of design freedom*" and operationally as "*a set of design principles*" [154]. As a background to this definition, NAF writes: "*In general, the design freedom of designers is undesirable large. The idea of architecture is to take advantage of this. Therefore, architecture is defined as normative restriction of design freedom. This idea of consciously applying normative restriction of design freedom is the really new thing. It makes architecture a prescriptive notion; any descriptive interpretation is cogently rejected*".
- The *ArchiMate Foundation* defines enterprise architecture to be "*A coherent whole of principles, methods and models that are used in the design and realisation of an enterprise's organisational structure, business processes, information systems, and infrastructure*" [78].
- The current architecture definition of *Capgemini* is: "*An architecture is a set of principles, rules, standards and guidelines, expressing and visualizing a vision and implementing concepts, containing a mixture of style, engineering and construction principles*".
- A recent definition from the *Gartner Group* is: "*Enterprise architecture (EA) is the process of translating business vision and strategy into effective enterprise change by creating, communicating and improving the key principles and models that describe the enterprise's future state and enable its evolution.*"

The variety in these definitions does seem to indicate that the field of enterprise architecture is still in its infancy. At the same time, however, the wide spread attention of enterprise architecture does indicate that enterprises do feel a profound need to steer their development (including their business and IT portfolio), and that they are looking towards enterprise architecture as a means to fill this need.

3.4.2 Perspectives on the role of enterprise architecture

While the above definitions may seem to differ considerably, what all these definitions seem to have in common is a reference to *structure* and *relationships* combined with a reference to a set of governing *principles* that provide *guidance* and *support* for *directions* and *decisions*. *Enterprise architecture* focuses on shaping and governing the design of the future enterprise using principles to stipulate future direction and models to underpin and visualise future states. In our opinion, there are three important perspective on the role of an enterprise architecture:

A regulation-oriented perspective – which manifests itself as a prescriptive notion governing the design of an enterprise. When taking this perspective one will focus on principles, leading to rules, guidelines and standards, focusing the enterprise's design freedom in the direction of its success.

A design-oriented perspective – which emphasises the comprehensive and cohesive specification of an enterprise in all its facets, as a high level design. This perspective focuses on essential design decisions, as well as its core structures. When taking this perspective, one typically produces models that describe the design of actual systemic artefacts and their interrelations.

A patterns-oriented perspective – which focuses on the use of design patterns. This perspective forms a bridge between the regulative and the design perspectives. To meet the regulations set out in the regulative perspective, during design activities, suitable patterns can be applied.

The regulation and design-oriented perspectives correspond to the earlier mentioned *indicator* and *control* aspects of the dashboard paradigm as depicted in Figure 3.5, and are complementary to each other in that the regulation-oriented perspective accommodates for the need to steer and direct developments, while the second perspective supports the need to gain insight into an enterprise's design while also providing guidance to designers of enterprise systems.

Even though not many definitions of architecture explicitly refer to the *patterns-oriented*perspective, the role of patterns to capture and re-use design knowledge (such as the quality attributes that will result from using specific patterns) in the creation of architecture (be it for buildings, software or enterprises) is evident [5, 15, 44, 123].

3.4.3 Definition of enterprise architecture

Using these perspectives, we can now define what we regard as enterprise architecture:

> *A coherent set of descriptions, covering a regulations-oriented, design-oriented and patterns-oriented perspective on an enterprise, which provides indicators and controls that enable the informed governance of the enterprise's evolution and success.*

3.4.4 Views in enterprise architectures

In practice, an enterprise architecture covers several foci that blend together to form the enterprise architecture. Without attempting to provide an exhaustive list, some typical (example) views are:

- In a *business view* one would define the integrated structure of the overall business itself (in terms of organisation, people and processes and resources). Business architecture supports business change with a more holistic perspective. This approach is becoming more important with the move towards service-oriented architecture at the business level.

- In an *IT view* one would define and describe the structure and relationships of IT systems including the way IT supports the enterprise to achieve its business goals.
- A *governance view* would addresses the full range of governance, from business governance (how to manage overall business processes, both formal and informal) to organisational and systems governance and also IT systems management capabilities.
- A *security view* addresses the full range of security, from business and information security to IT security. It also addresses the required security for organisational and business-related services. It is often linked to governance aspects to address security management.

In Chapter 4, we will discuss several dimensions along which to identify additional views. In the next Section, the concept of view will be defined as being one of the key concepts of enterprise architecture.

3.5 Key concept of enterprise architecture

Enterprise architecture can help organisations and their transformation processes in successfully executing their strategy. As such, it acts as an active planning and steering instrument, which can be used in translating strategy to programmes and projects, and revolves around four main components: principles, models, views and frameworks. Organisational transformation processes, embodied in programmes and projects, can use the principles, models and views as a means of content based steering in the coherence of the solution. In this Section, we will explore the concepts of concerns, principles, models, views and frameworks.

3.5.1 Stakeholders and their concerns

An enterprise has many stakeholders. Future development of an enterprise is likely to impact on the interests of these stakeholders. In this Section we briefly survey some classes of stakeholders and their specific concerns. In this book, we use the definition of stakeholder and concern as provided in [60]. A *stakeholder* is an individual, team, or organisation (or classes thereof) with interest in, or concerns relative to, a system (such as an enterprise). *Concerns* are those interests, which pertain to the system's development, its operation or any other aspect that is critical or otherwise important to one or more stakeholders.

In making decisions about an enterprise's future directions, stakeholders want to obtain insight into the impact these directions will have on their concerns, and understand the risks involved in current and future initiatives. Even more, since present day enterprises are complex social systems of interrelated processes, people

and technology, stakeholders are keen on finding a way to harness this complexity when judging the impact on their concerns.

As discussed before, each type of stakeholder has its specific need for insight, control and overview. At the same time, they all want insight into the potential impact on the enterprise resulting from changes in its own strategy or its environment, and consequences of decisions about the enterprise's future directions. They also have the desire to communicate about these changes and impact. Communication will take place at enterprise level, business unit level, department level and project level depending on the responsibilities of the stakeholder involved in the communication. Below we briefly zoom in on the interests and concerns of three typical classes of stakeholders, and their needs on enterprise architecture.

3.5.2 Principles

An univocal understanding about what is of fundamental importance for the organisation is essential. This is represented by the term "principle". Even though no broadly accepted definition of principle exists yet, principles are generally regarded as constraints on the design space for enterprise engineers [98]. According to TOGAF [139], principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organisation sets about fulfilling its mission. The extensible Architecture Framework (xAF) defines a principle as "a generic (functional or constructional) requirement for a class of systems" [154], where a class of systems is e.g., all enterprise information systems, so not only for an individual system. According to Capgemini's integrated architecture framework (IAF), a principle is a statement of belief, approach or intent which directs the formulation of the architecture, and may refer to the current state or a desired future state [30, 45]. In this book we will primarily follow the xAF definition as it provides an operational way of steering business and/or IT.

According to TOGAF, "*a good set of principles will be founded in the beliefs and values of the organisation and expressed in language that the business understands and uses. Principles should be few in number, future oriented, and endorsed and championed by senior management. They provide a firm foundation for making architecture and planning decisions, framing policies, procedures, and standards, and supporting resolution of contradictory situations*" [139]. As discussed in [22], when considering the many different definitions of principles, three typical perspectives on principles can be discerned:

Principles as inherent laws – referring to properties of (classes of) a system that can be observed and validated. Examples are the law of gravity, relativity theory, law of requisite variety, etc.

Principles as imposed laws – referring to properties of (classes of) a system that can be validated. Examples are: traffic laws, societal laws, policies and regulations within organisations, such as *we opt for customer intimacy*, *we comply with privacy laws* and *business flexibility has precedence over efficiency*. Prin-

ciples as imposed laws typically address the concerns of stakeholders. Some of these concerns may actually be triggered by an *inherent law* which might have a negative impact on the system/enterprise being engineered.

Guidelines – are properties of (classes of) a system that are specific enough to provide guidance to operational behaviour to make it fit within the borders set out by imposed laws, possibly referring to the use of mechanisms. For example: "use your car's cruise control" is an advisable *guideline* to abide by, in an effort to obeying *imposed laws* concerning maximum speeds on roads, using the inbuilt mechanism of the car's cruise control.

In line with the definition of enterprise architecture used in this book, we will primarily use the last two perspectives on principles.

3.5.3 Models

In general, models are a purposeful abstraction of reality. More specifically, a model is defined as "any subject using a system A that is neither directly nor indirectly interacting with a system B, to obtain information about the system B, is using A as a model for B" [8]. In colloquial use in the context of enterprise engineering, the term model is equated to some graphical diagram. This colloquialism can be explained as most models used in software development, business process (re)engineering, etcetera, are graphical models. Models, however, do not necessarily have to be graphical.

As depicted in Figure 3.8, in general, three categories of systems can be distinguished: concrete systems, symbolic systems and conceptual systems [35], also leading to three main classes of models. A concrete model of a concrete system is called an *imitation* (e.g. a scale model of a car). A conceptual model of a concrete system is called a *conceptualisation* (e.g. a process model as the conceptualisation of processes). A concrete model of a conceptual system is called an *implementation* (e.g. a process as the implementation of a process model). A conceptual model of a conceptual system is called a *conversion* (e.g. the algebraic concept of a circle ($x^2 + y^2 = r^2$) is a conversion of the geometry of its concept). A symbolic model of a conceptual system is called a *formulation*, and is expressed in some formal language. A conceptual model of a symbolic system is called an *interpretation* and is the reverse of a *formulation*. A symbolic model of a symbolic system is called a *transformation* (e.g. the transformation from Morse code to Roman notation of letters).

In enterprise architecting, a multitude of graphical and non-graphical models are needed. The set of required models spans over multiple dimensions of focus, goals and purpose. Some examples are:

- differing levels of realisation: from conceptual via logical to physical;
- differing aspects of transformation: from contextual (*why*) via design (*where to*) to the actual transformations (*how*);

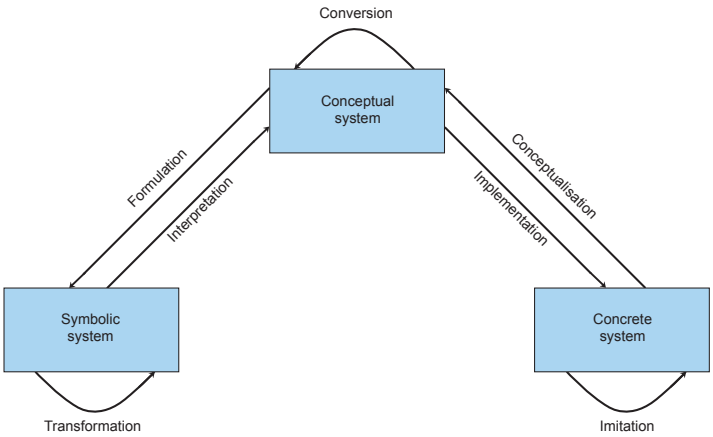

Fig. 3.8 Three types of systems

- different aspects of a enterprises: from goals via services, products and processes to IT;
- differing levels of aggregation: from enterprise level to the level of specific (partial) processes or applications.

Even more, models referring to one specific version / alternative of an enterprise, need to be coherent, also requiring coherence between models over the above dimensions. A core driver of the ArchiMate project [78] was also to increase the coherence between different aspects and models used in an enterprise architecture. In [78] several examples are shown which illustrate the need for coherence between different models used in an enterprise architecture.

3.5.4 Views

The complexity of the execution of an enterprise's strategy is likely to be immense because many processes, departments, and information systems are involved. When using enterprise architecture as a planning and steering instrument, then this instrument should reflect this complexity (the law of requisite variety [15]). As a result, it is almost undoable to make one single univocal and comprehensive set of models that can be used for all people concerned, therefore, several views are needed which focus on specific stakeholders and their concerns [78]. In Section 4.3, we will discuss the most common types of stakeholders involved in an architecture project. Stakeholders are important and their cooperation is necessary for a successful project, because they are the providers of resources, most of them are influencers, some even decision-makers, and they have information about objectives and constraints. Therefore, the architectural descriptions should answer their concerns.

Different views based upon the stakeholders concerns are an important communication means to obtain the cooperation of the stakeholders. A view is a representation of a whole system from the perspective of a related set of concerns [60]. This puts the notion of a view close to the notion of a model. We actually treat a model as being a special kind of view:

1. a model is a purposeful abstraction of reality that cannot be formally derived from an*other* model without changing the way in which the model represents the domain;
2. a view is a purposeful abstraction of reality that is derived formally from one or more models without changing the way in which the model represents the domain.

Therefore, each model is a view, but not each view is a model. As a background to these definitions, we refer to [129] Stachowiak distinguishes between three different "*model features*":

1. The *mapping feature*, concerned with the fact that a model is based on an original (the modelled domain).
2. The *reduction feature*, which deals with the fact that a model reflects a relevant selection of an original's properties.
3. The *pragmatic feature*, which is concerned with the usability of the model as a placeholder for the original with respect to some purpose.

Creating a model means creating/adjusting the *mapping feature* of a specific model. In creating views, one makes changes to the reduction and pragmatic features, without changing the mapping feature. Changing the latter would lead to another model.

3.5.5 Frameworks

The (example) dimensions for models as discussed above, apply to views as well. Even more, in the case of views one typically feels the urge to introduce views that are tuned to the interests and cognitive abilities of stakeholders as well as the communication goal at hand [107, 108].

To provide architects with some structure to select views, architecture frameworks have been introduced. These frameworks intend to aid architects by providing an ontology, which uses different abstraction levels to map all kinds of information needed. Architecture frameworks position architecture results and enable diverse communication (stakeholders, detail). Often tools and best practices are included in the framework to support the work needed.

3.6 Benefits of enterprise architecture

In Section 3.3 we already discussed seven key applications for enterprise architecture: *situation description*, *strategic direction*, *gap analysis*, *tactical planning*, *operational planning*, *selection of partial solutions* and *solution architecture*, enabling informed governance. We will now revisit the issue of the benefits of enterprise architecture as an instrument for informed governance, where we aim to make the benefits of enterprise architecture more explicit.

Even though a thorough scientific evaluation of the benefits of enterprise architecture is still lacking, the case for enterprise architecture has indeed been made by several market watchers, practitioners and business visionaries. Drawing on their study of numerous companies worldwide, [118] show how constructing the right enterprise architecture enhances profitability and time to market, while it improves strategy execution. A similar line of reasoning is expressed as "*To keep the business from disintegrating, the concept of information systems architecture is becoming less of an option and more of a necessity for establishing some order and control in the investment of information system resources*" in [155]. Nevertheless, an initial attempt for such evaluations has been reported in [121], though we still find objective figures lacking.

3.6.1 Uses of architectural descriptions

It goes without saying that enterprise architecture is a means to an end. This justifiably raises the question of the benefit of enterprise architecture. We position it to be a tool or means to support strategy formulation, planning and strategy execution. In essence an enterprise architecture is a tool to manage complexity and risks. It enables informed decision-making, planning and governing of transformations. As a means it can be used:

- within strategic business/IT planning;
- to align strategic objectives and IT;
- to define and guide large scale business and/or IT transformation;
- o structure organisation re-engineering;
- to enable design of organisational networks (shared service centres, BPO, etc.);
- to define and monitor IT programmes.

The IEEE working group on (software) architecture [60] mentions the following potential uses for architecture-models and associated descriptions:

- analysis of alternative architectures;
- business planning for transition from a legacy architecture to a new architecture;
- communications among organisations involved in the development, production, fielding, operation, and maintenance of a system;
- communications between acquirers and developers as a part of contract negotiations;

- criteria for certifying conformance of implementations to the architecture;
- as development and maintenance documentation, including material for reuse repositories and training materials;
- input to subsequent system design and development activities;
- input to system generation and analysis tools;
- operational and infrastructure support; configuration management and repair; redesign and maintenance of systems, subsystems, and components;
- planning and budget support;
- preparation of acquisition documents (e.g., requests for proposal and statements of work);
- review, analysis, and evaluation of the system across the life cycle;
- specification for a group of systems sharing a common set of features, (e.g., product lines).

Even though the IEEE working group was primarily working on software architecture, the above list of uses equally well applies to descriptions produced in the case of enterprise architecture (when replacing *system* by *enterprise* in the above texts).

In [13], the Software Engineering Institute has identified the following potential uses for architectural descriptions, which can also be generalised to enterprise architecture:

- it is a vehicle for communication among stakeholders;
- it captures early design decisions, both functional aspects as well as quality aspects;
- the global structure decided upon in the architecture, also structures further development;
- it is a transferable abstraction of a system.

3.6.2 Value of enterprise architecture

In terms of the uses as sketched above, and taking the dashboard perspective into account, enterprise architecture can deliver value to the business in many different ways. In an attempt to make this more concrete, the following are some examples of the values that can be realised through the use of architecture [30]. To demonstrate their impact effectively, they have been categorized as specific to Business, IT or both.

3.6.2.1 Value for the business stakeholder

- providing a full and coherent overview and understanding of an enterprise, i.e. people, roles, processes, organisation, goals, policies, rules, events, locations, etc.;

- providing an atlas and compass for management;
- business process improvement by structuring the business according to key services needed by the enterprise, based on a clear understanding of the goals/drivers of the business;
- eliminating (or resolving) enterprise duplication, enabling a move towards a "shared service" model, including identification of those services that may be better sourced externally (temporarily or permanently) [9, 10];
- underpinning decision-making on organisation splitting and organisation contracting [93, 94, 96];
- asses the impact of introducing a new product by determining whether the enterprise is able to deliver this product, which parts can be produced in house (by reusing current business services) and which parts should be outsourced (or produced by using external business services) [78];
- identifying opportunities for in-sourcing, including its consequences;
- a means to ensuring business compliance and governance;
- translating strategy in executable projects.

3.6.2.2 Value for IT

- reducing solution delivery time and development costs by maximizing reuse of models and existing systems, services and solutions;
- by conscious choices in abstraction, solutions can be designed that are either more agile for the same costs or consciously limited in their agility at a lower cost;
- reducing the risk of IT non-compliance with key regulations, especially as business becomes more regulated, e.g. Sarbanes-Oxley, etc.;
- ensure effective IT planning and management of IT roadmaps (and portfolio management), also enabling improved planning for resource skills and training and including application portfolio rationalisation [94];
- implementing and managing security by design instead of reacting to breaches as they are discovered;
- delivering solutions against IT Service Level definitions that are linked back to real business objectives and reduce instances of costly, ill-engineered solutions.

3.6.2.3 Value for Business and IT

- improving Business and IT alignment, allowing, for example, the identification of misalignment of individual projects with strategic outcome in early stages;
- ensuring alignment of data and information management with business objectives (e.g. partnerships);
- creating and maintaining a common vision of the future that is shared by both the Business and IT communities;

- ensure effective integrated change planning, reckoning with business and IT coherence.

3.6.3 Added value over classical approaches

It is a fair question to ask what the added value is of an architectural approach in comparison to existing approaches in which an enterprise's strategy is translated to a program of activities, which is consequently executed. Enterprise architecture is positioned (see Figure 3.4) between strategy and transformation program. This immediately raises questions such as: *What is new about this? Can't one do without it?* In Section 2.5 we already surveyed traditional approaches in meeting an enterprise's challenge. Now we have defined what enterprise architecture is, we can identify the added value over classical approaches:

- By designing a coherent conceptualisation of a solution first, one assures that programmes to realize the solution are complementary to each other instead of overlapping or even incompatible;
- It enables the management to more fundamentally and explicitly underpin their decisions about the sequence of projects;
- It offers guidance and boundaries for the realisation. When not applying architecture, each project will use the solution alternatives that are optimal for the project and possibly not the best for the coherent solution.

As also illustrated in Figure 3.4, enterprise architecture does not aim to replace the classic (mostly programme management) approach, but rather aims to complement it. Programme management and enterprise architecture need each other. Programme management cares for effectiveness and the control of time and budget. Enterprise architecture focuses on steering towards coherent solutions, aligning projects with this coherent solution, as well as setting boundaries for and providing guidance to the implementation of complex systems.

Whenever an enterprise is faced with a complex or messy problem [117] about its future organisational structure, IT support, etc, it is sensible to use architecture to gain better insight into the issues involved. By handling problems one by one, solution development becomes phased and manageable. The drawback is that solutions not necessarily match and fit together. In recent years, we have seen many examples of these mismatches in change portfolios. This yields surprises in systems management costs and users not being satisfied with their business and IT support.

By applying architecture, we treat problems in coherence. Instead of jumping to solutions right away, we develop a solution concept that takes away a fair amount of the degrees of freedom one traditionally used to have, but caters for detailed decisions made during systems development, while maintaining consistency. This sparks of interoperability, economies of scale (through common use) and possibilities for standards. It enforces a more consistent overall experience of the systems.

3.6.4 Use it wisely

After discussing the potential added value in the previous Subsection, one might wonder "*is architecture the cure for every issue an enterprise has to deal with?*" The simple answer is *no*. However, we will now add some nuance to this.

We do regard enterprise architecture as being a powerful means for management to obtain a holistic view of the enterprise and for facilitating decision-making and to set boundaries and provide guidance for implementation of complex systems. However, it is only a powerful means when it is applied properly and for the right reasons. So be aware when applying enterprise architecture that the chosen means does indeed fit the intended objectives. For example:

- Enterprise architecture typically provides management with an outlook on the coming three to five years. This outlook is rendered out of the many and various inputs the management provided them selves. Based on this outlook, management is able to plan programmes for the realisation of the chosen future direction. If an enterprise only aims to build a detached system with the intention to dispose of it, since it has no role of importance to play in the longer term strategy of the enterprise, one should not use enterprise architecture as a means to guide the realisation of this temporary system.
- At the same time, however, one needs to beware that these disposable, short-term solutions, do not actually become permanent or even worse, become critical for the execution of the business. When such a risk does exist, either an enterprise architecture should be used after all, or measures (governance!) should be taken to ensure that the disposable system is indeed disposed of.
- Enterprise architecture is used to provide insight and to reduce risks. If a system being designed is relatively simple and risk-free, applying enterprise architecture will not provide additional insights and is, therefore, overkill.

Some additional (anonymised) examples of scenarios leading to potential failures in using enterprise architecture are:

- To use an enterprise architecture for a different means than it was developed for. As a typical example: suppose a specific enterprise architecture was developed to support the development of a business case. By nature the business case will focus on feasibility of the proposed initiative and its cost/benefit. In this case, the enterprise architecture was designed as a high level solution and will help to find the major investment areas; as such it will not contain any guidelines for implementation. If the objective of this enterprise architecture is not clearly stated, an organisation may be tempted to use this enterprise architecture to guide realisation projects.
- An enterprise architecture is used after its period of validity. In some cases an enterprise architecture has been developed, but not put into use or not been maintained. If that same enterprise architecture is again used after some time, without checking whether it is still valid, the wrong decisions will be made.

- An enterprise architecture has been designed for a regulatory use, but no measures have been implemented to monitor and control the adherence of projects to these regulations.
- A view on the enterprise architecture that was developed to communicate about the enterprise architecture to senior management, is taken to be the actual enterprise architecture (and not the underlying models).
- Any initiative – including quick wins or those to solve an immediate issue – can only be realised if they adhere to the enterprise architecture. In this case the enterprise architecture, intended for strategic and long term initiatives, is wrongly used to restrict and complicate short term projects.
- Key decisions were made individually instead of in shared agreement with all relevant stakeholders, resulting in suboptimal and possibly overlapping solutions. Examples of this can easily be found in situations of decentralised governance, where for the same problem several solutions exist in several organisational units or (geographically defined) regions.
- Only a part of the architectural engagement was carried out. If, for instance, an aspect such as security is forgotten, it might lead to a coherent, but unsafe solution.
- One looks only from a limited perspective at the system being designed (not holistic). This might work well if it is by chance the most relevant perspective. Examples of this can be found where the IT department decides to implement new technology that does not effectively facilitate the business processes, most likely resulting in additional work in the business process as well as day-to-day irritations of the users.
- Enterprise architecture facilitates decision-making processes by providing a holistic view of the enterprise, leading to better-informed decision-making. At the same time, though, this is likely to make the decision-making process harder. This provides a challenge for the use of effective viewpoints providing decision makers with effective insight into all (and precisely all) relevant aspects affected by the decision to be made. When not using these mechanisms wisely, however, the result might be more confusion rather than more insight.
- Architecture is a means, and should not become a goal itself. One should only design the architecture at such level needed for the necessary insight and then stop. Nevertheless, numerous projects show that this risk is quite real!

3.7 Competencies of an enterprise architect

Even though Chapter 6 we provides a more detailed discussion of the skills required by architects and the challenges facing them, the discussions in this Chapter already do allow us to briefly reflect on these competencies.

As mentioned before, for a proper execution of a strategy, an additional means in addition to vision, strategy and programme management is needed: enterprise architecture. The means should be an unambiguous and understandable instrument sup-

porting stakeholders in their joint decision making, setting the direction and guiding the execution. Enterprise architecture should indeed reflect the shared conceptualisation of all stakeholders at a sufficiently specific level.

The enterprise architect will face the challenge of creating and applying such an instrument in a qualitative manner. For instance his architectural description should answer the concerns of the stakeholders. To what extent is it complete with respect to stakeholders, concerns and answers and to what extent is completeness feasible? In short: when is the architecture good enough from the perspective of product and process? Such an instrument should have a continuous value in steering the enterprise. Therefore enterprise architecture needs to be embedded in the overall change and governance processes of the enterprise, as is the case in portfolio and programme management. And it should be adapted to changes in the technology and business environment and concerns of stakeholders.

There is not one way of creating an enterprise architecture. Each specific situation has its own stakeholders, complexity, subject matter and scale. The current state of the craft is that many methods, tools and frameworks exist, both for products and processes. The quality of the enterprise architect determines the proper selection, adaptation and use of those in the specific situation. The current body of knowledge mainly exists of unrelated best practices in methods and frameworks. Current scientific developments work towards streamlining and finding common ground under successful best practices. We see enterprise architecture emerging as a new and exciting trans-discipline.

3.8 Summary

In this Chapter we started out with historical account of the term “architecture” and how its found its way from construction via computer hardware to software, and finally to the design of enterprises. To more theoretically underpin the role of enterprise architecture as an instrument for governance, we continued with a discussion of the *governance paradigm*. We proposed to regard enterprise architecting as a process involving a dashboard giving stakeholders *indicators* and *controls* allowing the gain insight into the current state of enterprise, alternatives for the future, as well as the performance of the transformation process(es), and to steer/direct these transformations. As a next step, we discussed seven key applications for enterprise architecture: *situation description*, *strategic direction*, *gap analysis*, *tactical planning*, *operational planning*, *selection of partial solutions* and *solution architecture*, enabling informed governance.

We then went on to discuss several definitions of architectures, finally leading to an understanding of how enterprise architecture is regarded in this book. In our definition of enterprise architecture, we regard it as being combination of a *regulation-oriented*, *design-oriented* and a *patterns-oriented* perspective, where the design-oriented perspective is mainly pivoted towards the indicators on the dashboard and the design-oriented perspective towards the controls.

We then turned our focus to the key concepts of enterprise architecture: stakeholders, concerns, principles, models, views and frameworks. Using the discussion of the key concepts as a background, we then revisited the potential benefits of enterprise architecture, and the potential value of architectural descriptions. We also stressed the fact that enterprise architecture is not the right answer to all situations. When applied in the wrong situation, serious negative consequence may result. We finished this Chapter with a first exploration of the challenges that should be met by enterprise architects.

3.9 Discussion statements

1. Enterprise architecture is answering the needs, that business administration and organisational design should address! Even more, we should add this knowledge to the body of knowledge for business administration and organisational design.
2. Enterprise architecture should restrain itself to the regulation-oriented perspective; all other work is simply high-level design. There is no need for yet another concept.
3. Enterprise architecture is an enjoyable intellectual exercise. However, in the end, senior management will take their decisions mainly based on their business feeling and intuition.
4. Full Business/IT alignment can not be reached without the use of enterprise architecture.
5. The instruments of the enterprise architecture (principles, models, views, frameworks) are far less important than the process involved-in/concerning enterprise architecture.
6. The creation and use of an enterprise architecture takes too long, and therefore falls in exactly the same pitfall as “endless strategy formulation”.
7. A project which aims to realise a part of an enterprise architecture does not need an enterprise architect. The enterprise architecture itself provides all the guidance needed.
8. Architecture principles are useless since they are not specific enough. A real design is much more useful as it makes the final result much more tangible.

Chapter 4
The Results of Enterprise Architecting

4.1 Introduction

During the process of enterprise architecting several results can be produced. These results are not limited to principles, models (including their cross-references) and views alone. Other results are for example intermediate results used to develop the enterprise architecture and the evaluation of alternative solutions/directions. Some of the important results of an enterprise architecting effort do not even have to be tangible, for example shared understanding, shared agreement, and commitment amongst stakeholders.

The actual set of results that should be produced, and the required level of detail and form of the results, depends on the stakeholders and their concerns, as well as the decisions that should be taken based on these results. Even in a simple example it becomes clear that enterprise architecture can produce a large variety of tangible and intangible results. Therefore criteria on product quality are needed to make choices which results should be delivered. In such choices we can also profit from an insight how the possible results interrelate, as described in architecture frameworks.

The remainder of this Chapter is structured as follows. As a simple basis for the whole Chapter, we will start providing examples of enterprise architecture results, using the relatively simple case enterprise: *Pizzeria "Perla del Nord"*. Then we will reflect on product quality and its criteria: when is good *good enough*? Subsequently we will revisit architecture frameworks as a means to structuring tangible enterprise architecture results. This is followed by a discussion of a methodical perspective on the creation of models and views used in enterprise architecture. Before concluding, we will also argue the point for a language with a unified look and feel for the representation of architectural models.

4.2 Example enterprise architecture: Pizzeria "Perla del Nord"

In the remainder of this Chapter we want to be able to regularly refer to specific enterprise architecture results. We therefore introduce a running example: *Pizzeria "Perla del Nord"*. This running example is primarily inspired by [34], while the principles used in the example are derived from [36]. It should be noted that the Pizzeria case is a small case, and is not intended as being exemplary for enterprise architecture cases in real life. The purpose of the Pizzeria case, however, is to illustrate the workings of some key elements in enterprise architecture, and not to mimic a real-life enterprise architecture. The latter is beyond the goals of this introductory book.

For *Pizzeria "Perla del Nord",*we will discuss its current objectives, its design principles and some models reporting on its current design. We meet the pizzeria just at the moment they intend to play a significant role in the business-to-business (B2B-)market. This is the result of a change of strategy, in the sense that the pizzeria intends to open up a new market (business-to-business). For this intention we will subsequently show the impact-of-change, as required by the CEO of Perla del Nord to govern the intended transformation. As the CEO's desires are the primary driver for the creation of this architecture, the focus of the example will be on the business aspects.

4.2.1 Current situation

The pizzeria is located in the city centre of a medium sized town. In and around this city centre a large number of offices are located. The mission of the pizzeria is *to offer positive influence in the work-life balance of both yuppies and dinkies*. The desired future state is *to be the most renowned pizzeria in the region* and *be the number one customer preference* which will be accomplished by *organic growth* and *sharing the profit with employees*. This vision is made more specific in terms of a number of goals (*grow from 100 pizzas a day to 150 pizzas per day in the next three years, improve profitability by 10% per year and to improve quality, measured by diminishing customer complaints*) and policies (*management development* and *financing programme for talented employees to support them in starting their own branch*).

In the current situation, the pizzeria is designed according to a number of business principles:

1. *Reward valued customers*
2. *Outside own organisation: get money first*
3. *Bake to order (not to stock)*
4. *Bake while driving*

No explicit technical infrastructure principles have been used in the current organisational design of the pizzeria. In discussing the pizzeria case, we will consider the pizzeria at three levels of abstraction (leading to a clear separation of concerns):

Valuation – The value exchange between the pizzeria and its environment, focusing on the value each actor potentially adds to the (economic) goals of other actors.

Function – The functionality provided by the pizzeria to its environment in terms of business services, transactions and their orchestration. In the case of the pizzeria, the business services will be services that are offered to clients, in line with the business goals identified in the pizzeria's strategy.

Construction – The construction of the pizzeria in terms of sub-actors.

The distinction between these levels is motivated as follows:

- Depending on the actors, and their goals, a business service (such as *deliver pizza*) may provide different values to different actors. This motivates the distinction between the valuation and function layers.
- A given business service may be realised using different organisational structures. In other words, for a given business service, several alternative constructions can be used for their realisation. This provides the motivation for a distinction between the function and construction layers.

In modelling the current design of the pizzeria at these three levels, we use three different notational styles:

1. e3Value [46],
2. DEMO [114] and
3. ArchiMate [78].

The highest-level view of the pizzeria, the *valuation* level, is shown in Figure 4.1 in terms of an e3-value diagram. *Customers* and the *Pizzeria* may engage in a value exchange where money flows to the *Pizzeria* in exchange for a pizza. *Customers* are regarded as the initiators of this value exchange.

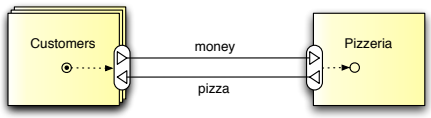

Fig. 4.1 Pizzeria at the valuation level

At the next level of abstraction, we are interested in the functionality provided by the Pizzeria to its environment. For this situation we assume that the money/pizza value exchange is to be realized in terms of two business services, namely *complete purchase* and *pay purchase*. Below we will see how moving to a B2B context will lead to a third business service. The diagram depicted in Figure 4.2 is a DEMO construction model, expressing the coherence (chain/network) of business services,

delivered by actors to other actors within a defined scope. In DEMO each business service is delivered by a transaction, consisting of a production embedded in four standard phases of co-ordination, as follows:

Request – An actor requests a business service from another actor.
Promise – The requested actor promises to deliver the business service.
Now the requested actor produces the requested result, i.e. the service is executed.
State – The requested actor states/claims it has indeed delivered the service.
Accept – The requesting actor accepts the result of the business service.

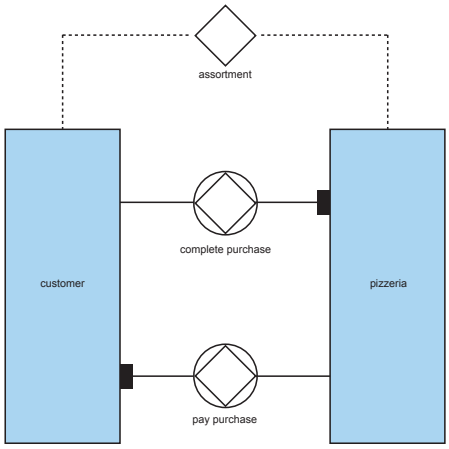

Fig. 4.2 Transactional view of value exchange

The transaction symbol combines a circle, signifying the four coordination phases, and a 45° rotated square, signifying the production. DEMO also discerns several exception handling procedures since there are likely to be numerous ways in which a transaction can break down.

When considering the two transactions between a customer and the pizzeria, which are involved in a pizza-for-money value exchange, there are several ways to interleave the four phases of the transactions. Should a pizza be paid before it is delivered? Or even before the pizzeria promises it to be produced? A selection between these options should ultimately be motivated in terms of the pizzeria's strategy.

In Figure 4.3 we have depicted the current orchestration of the two transactions. In this orchestration, the *customer* places an order (e.g. by phone) and immediately pays for it (e.g. using a credit card). The pizzeria responds to the request for purchase by a request for payment, which is required to be followed by the actual payment (promise / state / accept), after which the pizzeria promises to deliver the pizza (and indeed will do so).

When we decompose the construction of the pizzeria, this leads to the top-level construction as shown in Figure 4.4. We now see the chain of dependencies within

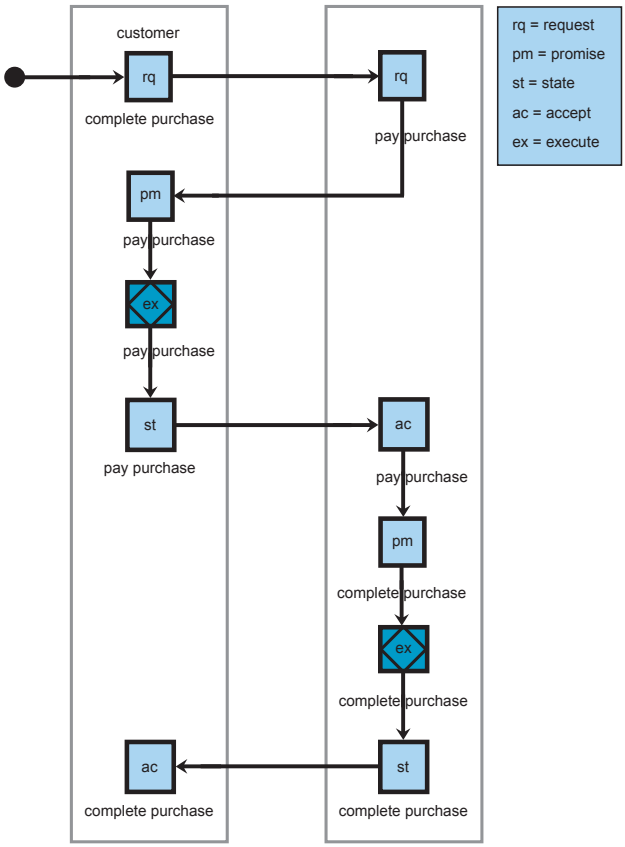

Fig. 4.3 Orchestration of transactions

the pizzeria: it is actually the *completer* who provides the business service *complete purchase* and uses the business service *pay purchase*. What is not shown in this diagram, is the fact that in executing *pay purchase*, the *completer* will offer customers a *discount* based on their order history. The *completer* in turn depends on the *baker* for the *baking of the purchase* and a *deliverer* for physically *delivering the purchase*. Also extra business resources appear now: the baker of course refers to the same assortment, but also uses *recipes*, the completer uses *customer data* and the *deliverer* uses *maps*.

The decomposition of the pizzeria’s construction also allows for a refinement of the transaction orchestration. This is shown in Figure 4.5. After the payment by the customer has been executed, the *completer* confirms the order to the *customer* and orders the *baker* and the *deliverer* to bake and deliver the order to the *customer*. Baking and delivering the order are executed in parallel: the *baker* prepares the pizza and places it in the oven on the back of the moped. Finally the order is delivered to the *customer*.

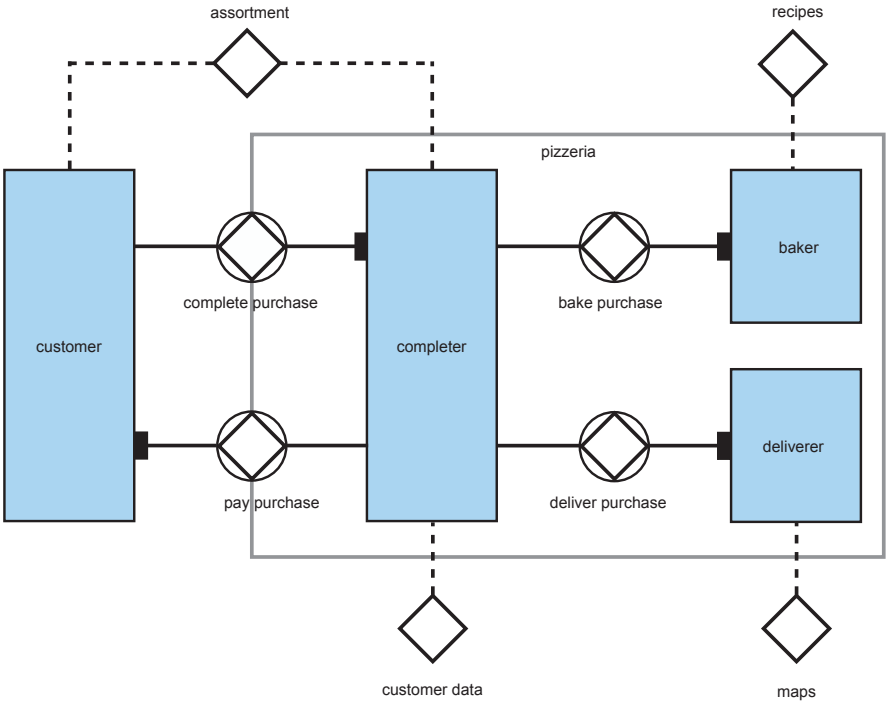

Fig. 4.4 Decomposition of the Pizzeria

Looking at the current situation in terms of principles, we see that the principles have been implemented as follows:

- *Reward valued customers.* Returning customers get a discount based on their purchase history.
- *Outside own organisation: get money first.* The order has to be paid before it is confirmed.
- *Bake to order (not to stock).* The order is baked after the customer has ordered.
- *Bake while driving.* Baking the order and delivering the order are executed in parallel by using mobile baking technology.

Figure 4.6 provides an overview of the orchestration of transactions including actor roles responsible for each phase in the transaction. Actor roles are implemented by *functionary types*, being a cluster of responsibilities for co-ordination and production acts, such that a person can be assigned to that cluster.

As a general rule, the functionary type who performs the promise in a transaction is considered to be the one who is authorized to be the executor of all phases of that transaction. But due to obvious reasons, the functionary that promises the transaction *complete purchase* (order taker) will not physically go to each *customer* to state the *complete purchase* transaction. This authority is, in our case, delegated

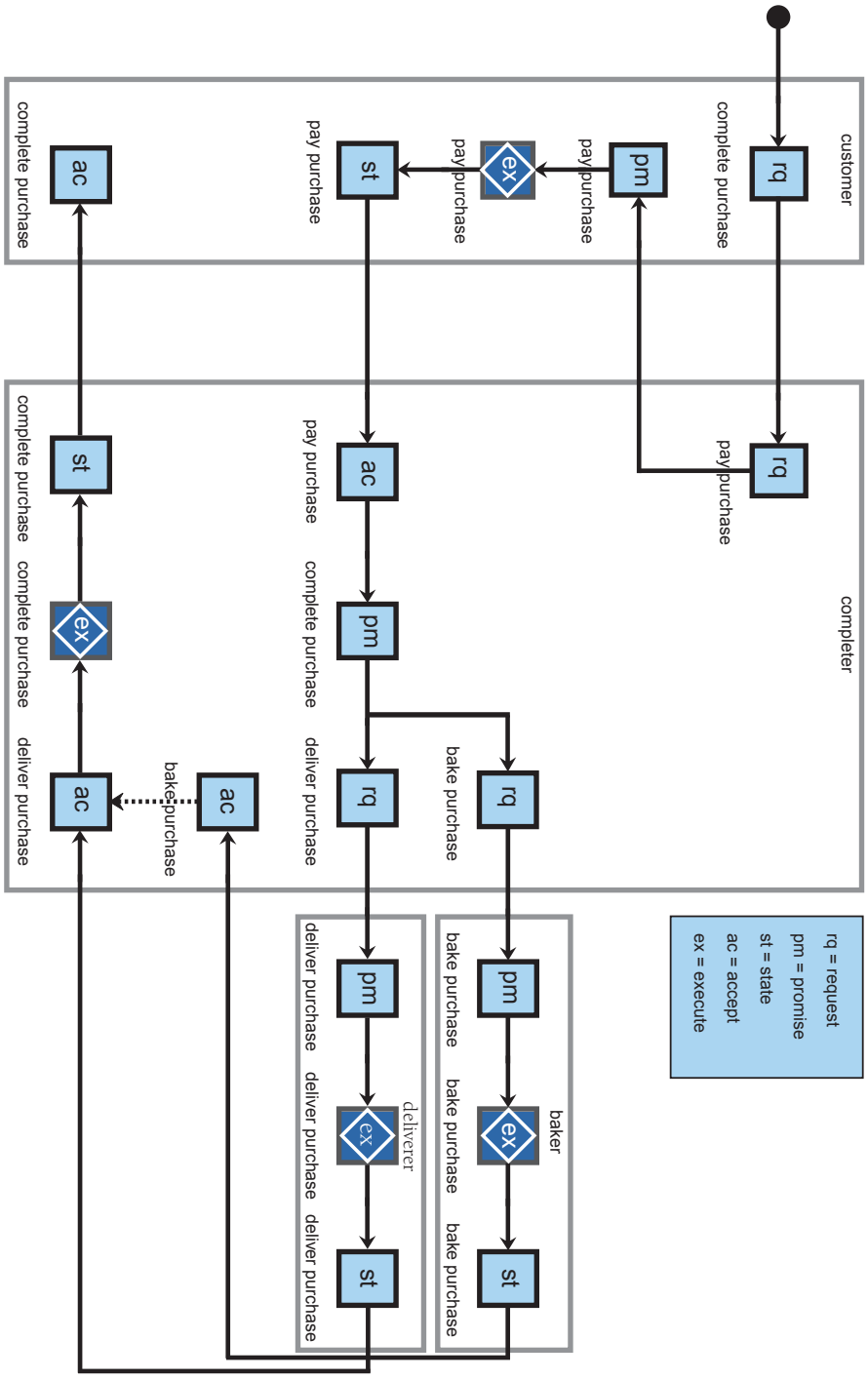

Fig. 4.5 Composed transaction orchestration

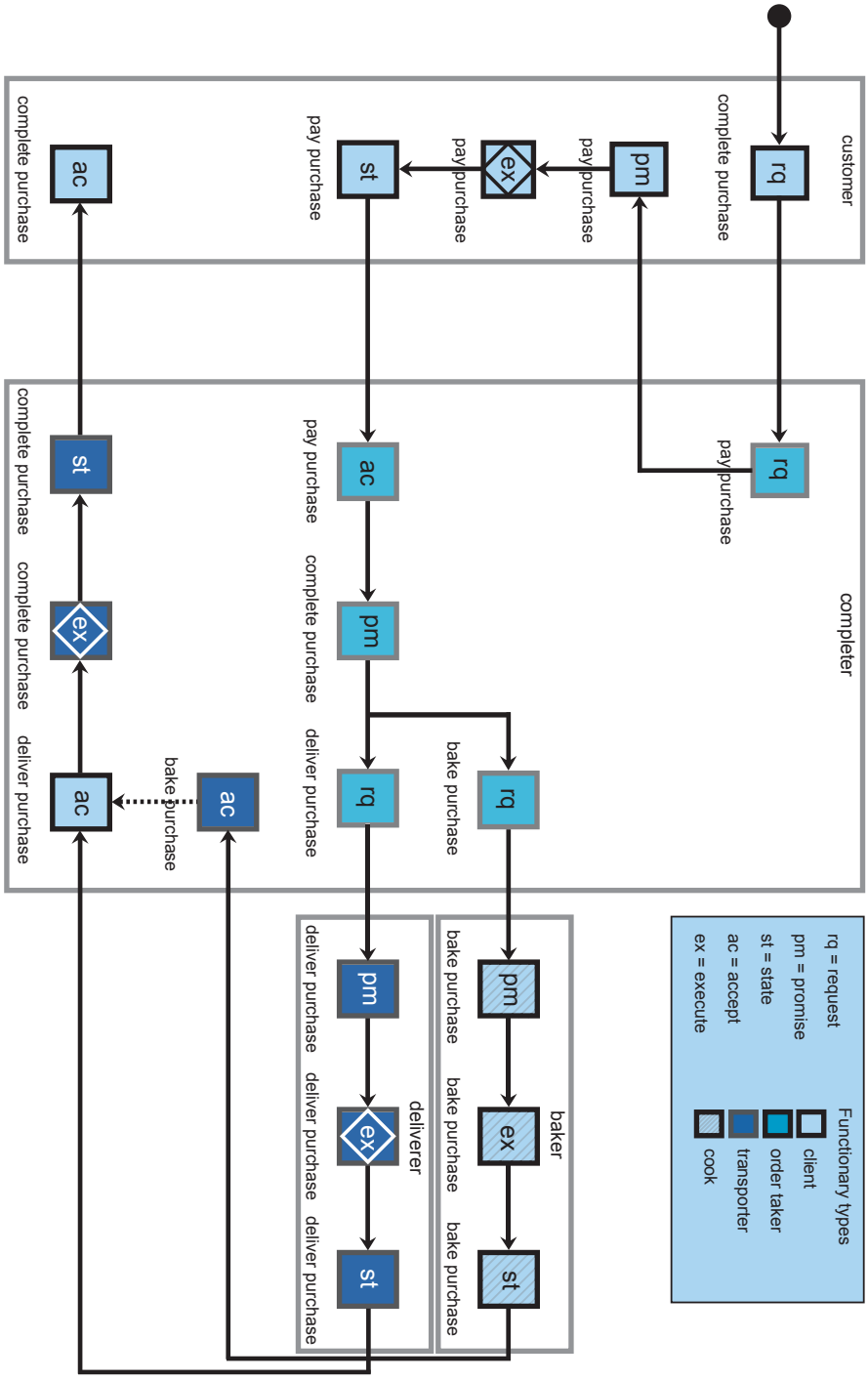

Fig. 4.6 Composed transaction orchestration including functionary types

to the functionary type *transporter*. Delegation does not transfer the responsibility: the order taker remains responsible for the entire transaction.

To support their operations, the pizzeria uses several application services and technical infrastructure services: a website which contains the assortment customers can choose from, electronic map software to plan the optimal route to the customer, accounting software, an external hosting service for the website, a PC for running the accounting software, a GPS navigation device and a credit card terminal. This has been illustrated in Figure 4.7, using the ArchiMate notation.

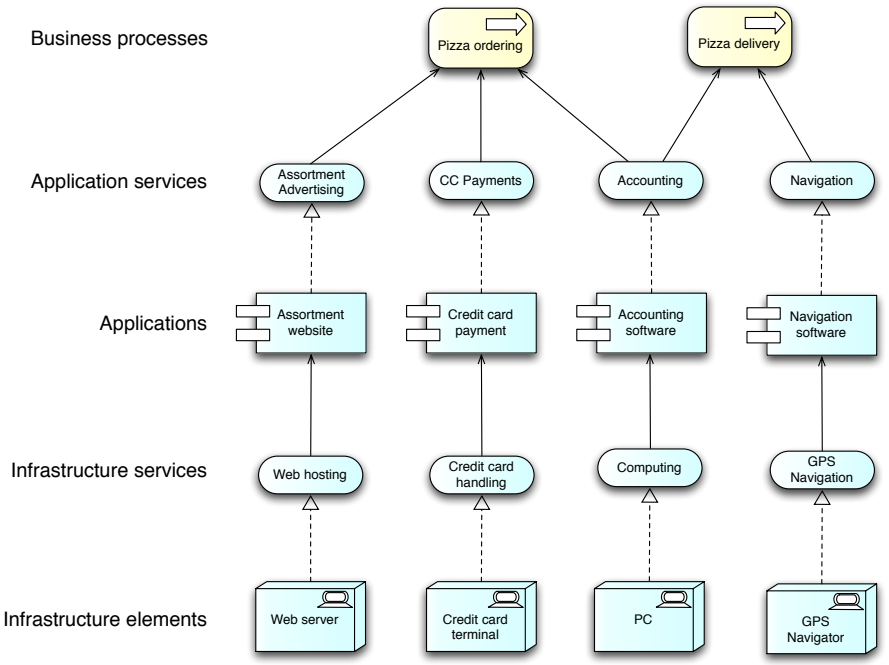

Fig. 4.7 The current software and technical infrastructure services

4.2.2 Intended change

Management of the pizzeria “Perla del Nord” has decided to enter the business-to-business (B2B) market within the next three months, primarily to be able to provide pizza delivery services to offices in and around the city centre. This will enable them to realize the intended growth from 100 pizzas a day to over 150 a day. As an additional service, they will allow B2B customers to pay afterwards by sending them monthly invoices. Before B2B customers are allowed to pay afterwards, their

credit rating is checked first (only B2B customers with a positive rating are allowed to use this service) and a contract must be signed. The pizzeria will use electronic 3rd party services for the credit rating, because they want to dedicate their resources to their core business.

Since the intended change will have its impact on the core processes of the pizzeria, the CEO decides to use an enterprise architecture for the following purposes:

- to speed up the decision making process of the intended change;
- to outline operational and transformational consequences of the intended change in terms of organisational change, human resource policy (e.g. required number of employees, competences), costs and benefits, new business partners. An operational consequence of the intended change is for instance new business transaction for contracting. Transformational consequences are for instance developing a contract template for B2B customers and contracting 3rd party services;
- to determine the best implementation scenario;
- to communicate the intended change to different types of stakeholders: e.g. a view for the CFO which addresses the costs and the necessary additional software or a view for the HR manager that addresses the new competences.

The enterprise architecture, of the desired situation, should be in line with the new strategy. In doing so, the first step is to determine the impact of the principles on the intention of entering the B2B market. Next the impact of the intended change on the construction or the pizzeria organisation is defined.

By entering the B2B market the principle of *Outside own organisation: get money first* needs to be amended to: *Outside own organisation: without a contract, get money first*. New principles are introduced as well: *Focus on core business: baking and delivering pizzas* (business principle), *only control data from own core processes* (information principle) and *prefer using controlled data collection from 3rd parties* (information principle). Since the pizzeria is going to exchange information with 3rd parties, they open their network to the external world. They want to realise this in a secure way leading to a technical infrastructure principle: *controlled access towards 3rd party networks determined at the level of the enterprise service bus (ESB)*.

At the construction level, as illustrated in Figure 4.8, new actor roles are introduced: *B2B-contractant* and *credit rater*. It has to be decided whether this role stays within the organisation of the pizzeria or if this role is to be outsourced or shared. Note that there is no additional actor role introduced for invoicing B2B-customers. We see this as a specific process implementation of the transactions *complete purchase* and *pay purchase*: the request of *pay purchase* for B2B-customers is executed once a month instead of directly after the request *complete purchase*. We also see additional transactions appearing: *rate credibility* and *contract-B2B*. For the credit rating an additional business resource (fact bank) is needed. This fact bank is external, since the pizzeria prefers to use controlled data collection from 3rd parties. We also see that the money/pizza value exchange depicted in Figure 4.1 is now realised in terms of three business services, delivered by three transactions: *complete purchase*, *pay purchase* and *contract B2B*.

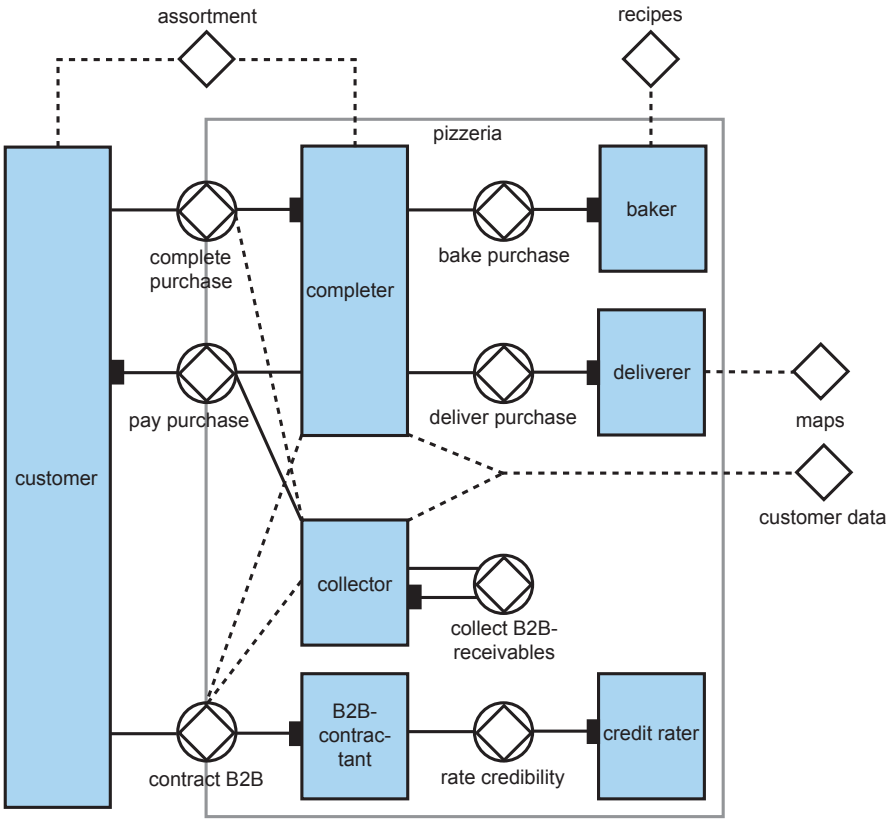

Fig. 4.8 Construction of desired situation

Entering the B2B market yields additional needs to for information and software services: a service to store data of B2B customers (such as contact and delivery information), a service to store the order history of B2B customers as input for e.g. invoices, a service to store the credit rating of B2B customers so the pizzeria does not have to check the credit rating each time the B2B puts an order in, and a service for checking the credit rating of B2B customers at a third party. Finally, a financial service for sending and monitoring the payments of invoices to B2B customers is necessary. The first three services are implemented in a CRM system. The financial service is implemented by using the existing accounting software. For the last service, an external Web service is used.

Entering the B2B market also yields additional needs with regards to technical infrastructure services: a server to run the CRM software (this server will also host the accounting software), an ESB for securely connecting to the credit rating web-service, a local area network to connect the server to the ESB and the PC and a fire-

wall to secure the connection to the Internet. The result is illustrated in Figure 4.9.

To demonstrate the impact of the intended change on the activities of the (functionary type) order taker, the artist impression shown in Figure 4.10 was used in which the changes impacting the work of the order taker have been highlighted. In the new situation the order taker needs to check whether the customer is indeed employed by a company that has a contract with Perla del Nord. Therefore, customers are required to identify themselves using their (company issued) access card and pin number. The order taker must have access to the administration of B2B contracts.

4.2.3 Enterprise architecture results in the Pizzeria case

To finalise the pizzeria case we briefly discuss a non-exhaustive list of specific enterprise architecture results, which can be derived form the pizzeria example. In doing so, we will distinguish different types of deliverables:

- final deliverables;
- intermediary results;
- intangible results.

The first two result types, which are tangible results, are concerned with designing the architecture, communicating the architecture, performing analysis and drafting recommendations. Intangible results are mainly concerned with creating commitment and a shared understanding amongst stakeholders. The pizzeria example yields the following final deliverables:

Models – In this example the Construction Model of the DEMO method has been used. The model presents a univocal, comprehensive, and concrete image of the desired state. The model is an ontological representation of the pizzeria showing the boundaries of the system, the actor roles and the transactions between actor roles. This model can be used to determine the implementation of actor roles in organisations (including sourcing and sharing opportunities), the implementation of the process flow and to identify business services (based on the transactions) including the quality of business services (QoB). E.g. the QoB of the business-service pizza-delivery tells what the bakery will deliver, when the bakery delivers, how much pizzas the bakery delivers and how good the quality of the baked pizzas is. The model can also be used to identify information service, including the quality of information services (QoI). E.g. the QoI of the information service *determine discount* states that the information service must be real-time available, the actuality on the information must be less than 1 minute, meaning that purchases less than a minute ago will not be included in determining the discount.

Stakeholder-specific views – Views are necessary to provide stakeholders with insight in the potential impact on their concerns. E.g. a view for the completer to illustrate the changes in his responsibilities due to the expansion to the B2B

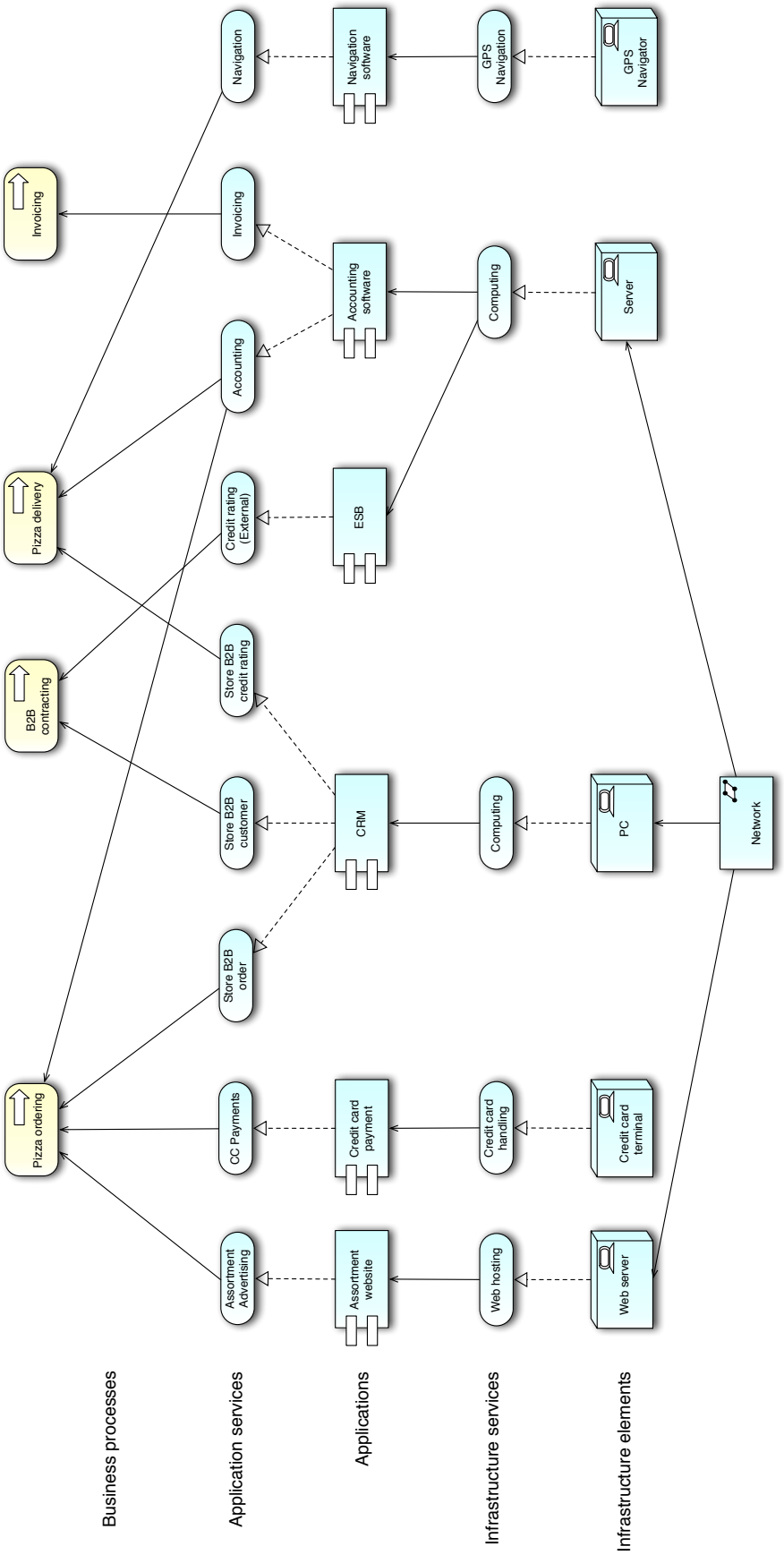

Fig. 4.9 Software and technical infrastructure services of the desired situation

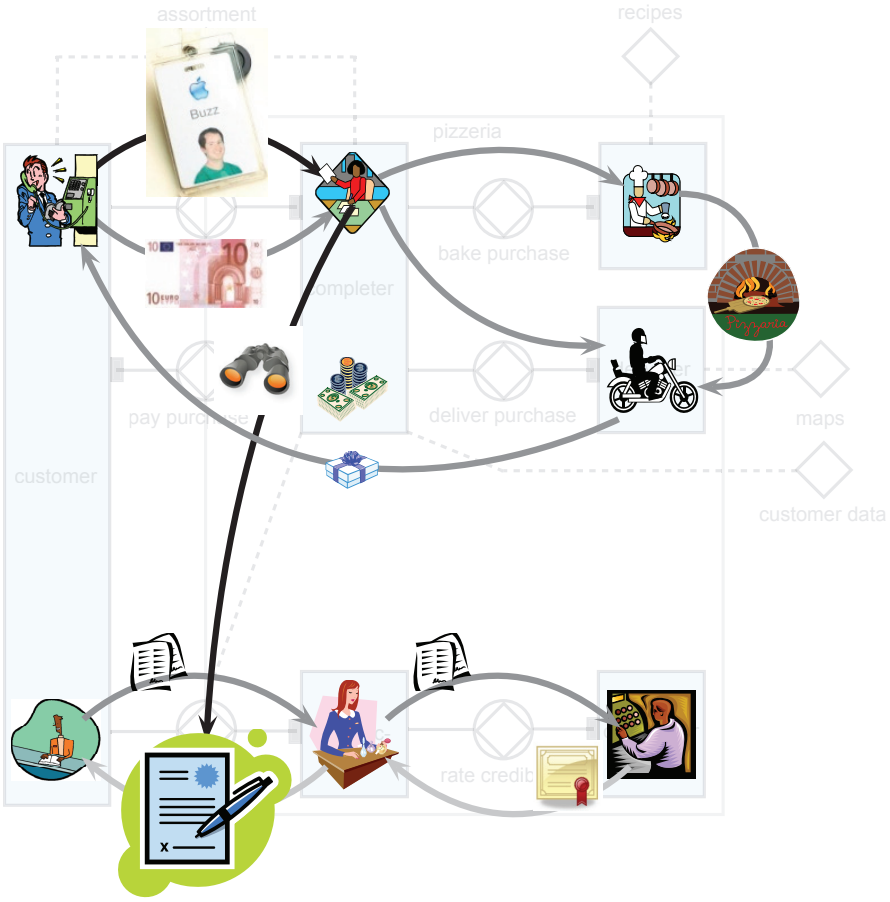

Fig. 4.10 Artist impression of new work environment of order taker

market, a view which shows all internal and external stakeholders or a view showing the connections with 3rd parties which can be used for determining the necessary security measures. Views can take any form, not only narrative descriptions or landscape maps, but also artist impressions, animations, simulations, role plays, games, etcetera.

Specifications and guidelines – The architecture can be used for several specifications such as:

- Determine process flows;
- Define responsibilities;
- Requirement specifications for software based on the QoB and QoI of business- and information services. E.g. the software system used for the

information service *determine discount* should be available real time and during service times with and actuality less than one minute;
- Security guidelines for access towards 3rd party networks.

In the pizzeria example we have seen how the following intermediate results have been produced:

Principles – Principles guided the design of the pizzeria. E.g. the business principle *Outside own organisation: without a contract, get money first* had its impact on the physical implementation of the process flow and the technical infrastructure principle *controlled access towards 3rd party networks determined at the level of the enterprise service bus (ESB)* determined the implementation of security with 3rd party networks.

Solution alternatives – During the design of the pizzeria, some solution alternatives have been identified, for instance a solution alternative where the business services *baking purchase* and *delivering purchase* are being outsourced or a solution alternative where information about the payment history of B2B customers is shared with the *credit-rater* in order to get a discount on their services. The enterprise architecture can be used to determine the validity and feasibility of each solution alternative in order to select the most appropriate alternative.

Traceability of decisions – E.g. the reason why baking and delivering pizzas are executed in parallel can be traced back to the principle of *bake while driving* and the process flow of the pizzeria can be traced back to all business principles.

In the pizzeria example the following intangible results would have played a role (if it were a real case):

Communications amongst involved organisations – The enterprise architecture can be used for communications among organisations involved in the system. E.g. communications between the pizzeria and the 3rd party delivering credibility data as part of contract negotiations: what is the QoI or data ownership;

Commitment amongst stakeholders – Stakeholder specific views, addressing their concerns, will increase their commitment for the intended change.

Even in this simple example of the pizzeria, one can see there are several different results enterprise architecture delivers. The challenge is to determine which products to deliver (when is good, good enough) and to safeguard consistency of interrelations between these products.

4.3 Quality of the produced results

A lot of resources, such as money, time, emotional energy, and intellectual energy are invested into the creation of enterprise architecture results. This raises the question of whether these resources are spent well. *Do the results meet the needs?* In other words: *What is the quality of the results? What is the return on modelling effort (RoME)?*

4.3.1 Possible uses of architecture results

In considering the quality of enterprise architecture products, we take the position that the stakeholders, their concerns, as well as their information, insight and/or steering requirements as a starting point. Some typical examples of such needs are:

Top-level management – Is the intended transformation still justified given the (expected) improvements in relation to the (expected) costs of the transformation? How can we ensure our policies are followed in the development and operation of processes and systems? What is the impact of decisions (on personnel, finance, IT, etc.)?

Middle-level management – The current situation with regard to the computerised support of a business process.

End user – The potential impact of a new system on the activities of a prospective user.

System administrators – The potential impact of a new system on the work of the system administrators that are to maintain the new system.

Operational manager – What new technologies do we need to prepare for? Is there a need to adapt maintenance processes? What is the impact of changes to existing applications? How secure are my systems?

Architect – The requirements of a specific stakeholder with regard to the desired situation. What are the consequences for the maintainability of a system with respect to corrective, preventive and adaptive maintenance?

Project manager – What are the relevant domains and their relations, what is the dependence of business processes on the applications to be built? What is their expected performance?

System developer – What are the modifications with respect to the current situation that need to be performed?In the ArchiMate project [78], three purposes for models and views have been identified:

Designing – supporting architects and designers in the design process from initial sketches to detailed design.

Deciding – supporting decision makers in the process of decision making by offering an insight into the issues/impacts they need to decide upon.

Informing – supporting the informing of any stakeholder about the enterprise architecture and its impact on the future enterprise.

In the ArchiMate project, the focus was on architecture level design models as well as the creation of associated views. Design principles and general business requirements were not taken into consideration. They are, however, also possible results produced in an enterprise architecting effort, albeit not of a design nature. Therefore, it is wiser to generalize the notion of a *designing* purpose to a *specifying* purpose, be it the specification of a design, a set of principles, or general business requirements. Furthermore, developments such as outsourcing stress the purpose of architecture results for contracting reasons. Architecture results have become part of formal contracts governing outsourcing and/or procurement [42]. In sum, we identify four types of goals for which architecture results may be created:

Specifying – making explicit the requirements, principles or designs pertaining to the enterprise, ranging from initial sketches to detailed specifications, including their justification in terms of earlier made decisions.

Deciding – supporting decision makers in the process of decision making by offering an insight into the issues/questions, as well as their consequences and possible justifications, they need to decide upon.

Informing – supporting the informing of any stakeholder about the enterprise architecture and its impact on the future enterprise, possibly including their justification.

Contracting – providing a formal statement of the architectural requirements of enterprise/system components to be realized/worked-out by a supplier.

4.3.2 From intended use to the design of a result

Given this spectrum of uses, the creation of an enterprise architecture results requires deliberate planning. It is not just a matter of coming up with the right content, but also selecting the right depth, subjects, forms, etcetera. In other words, they need careful designing as well. This "result design" involves the identification of the intended use of the result, the intended audience, the subject of the result, and the form to use in representing the result. The subject, and therefore the scope, of the planned result will be dictated by the concerns/interests of the intended audience, and may for example be: value exchanges in the value chain, business services offered, technological infrastructure, detailed process designs, process performance, requirements, principles, etc. The form the result takes refers to the communication style and languages used. One may, for example, opt for an intangible form or a tangible form such as a graphical model/view, a textual model/view, a combination of the latter, or even an animation. For the models one will typically have to make a selection from languages such as:

ArchiMate – a language [78] to express the (design oriented) architecture of enterprises.

DEMO – a language [35, 114] to express the ontology of an enterprise.

e3-Value – a language [46] to express the value exchange between business actors.

UML – even though it [25] is initially designed for software design, has been extended for business modelling [39].

Some of these languages provide pre-defined mechanisms to create views, for example, UML's swimming lanes. The ArchiMate project also defined a number of such mechanisms in terms of landscape maps, process illustrations, etcetera [78]. In addition, one may want to construct ad-hoc views for specific uses.

4.3.3 A deeper understanding of the quality of results

The quality of the results of enterprise architecting efforts is relative to its intended use. Some research has been conducted into the quality of models and the modelling process [24, 74, 75, 76]. Even though not all enterprise architecture results are models, these results can be applied more widely. One of the reasons for this is the fact the notion of *model* is used in a more general way in the research on quality of models than we have defined it in this book. For example, a set of design principles can also be regarded as a *model* representing the intended restriction of design freedom.

Figure 4.11 shows the Sequal framework for model quality as presented in [74]. In this framework a distinction is made between:

Physical quality – How the model is physically represented and available to stakeholders; a matter of *medium*.

Empirical quality – The *comprehensibility* of the model to its intended audience in terms of size, complexity, the number of symbols/graphemes used in a model, etc.

Syntactic quality – Conformity to the syntax of the modelling language.

Semantic quality – How well the model reflects the knowledge (harboured by the domain expert) of the modelled domain.

Social quality – The level of agreement between the stakeholders involved about the model.

Given an intended use of a planned enterprise architecture result, this can be translated to specific requirements with regards to each of the above aspects of model quality. Based on these requirements, an appropriate subject and form for the result can be selected, as well as an effective strategy to create the result [24, 57, 107].

4.4 Enterprise architecture frameworks

As mentioned in the pizzeria Perla del Nord case, apart from establishing which products to produce, it is also a major challenge to safeguard the consistency of interrelations between these products. Architecture frameworks have an important role to play in this. In our view, enterprise architecture frameworks provide:

- a means to order architecture results;
- a means to guard their completeness, both in terms of scoping and level of detail;
- insight into the interrelationships of architecture results, enabling the traceability of decisions and their impact.

In this Section, we therefore provide an exploration of some of these frameworks. The aim of this exploration is to illustrate the range of frameworks. In the next Section we will undertake an attempt to more structurally define the notion of

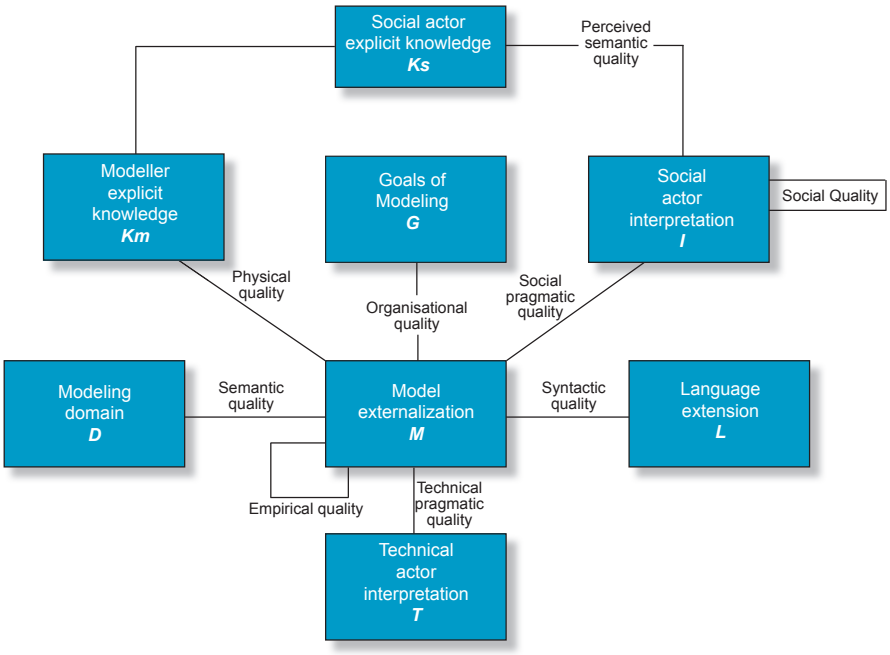

Fig. 4.11 The Sequal framework for model quality

framework in an enterprise architecture context and the possible dimensions these frameworks may occupy.

4.4.1 Tapscott & Caston's views

In [135], the framework as depicted in Figure 4.12 was presented as a way to position different interests with regards to an enterprise and work towards solutions which align these different interests. The focus of each of the views is defined as:

Business view – The *business view* highlights *what* business is conducted by the organisation (or organisational sub-unit), in other words *'the line of business'*. This view considers an organisation as a service providing entity. The business view aims to describe only *what* an organisation does in terms of the services it provides.

Information view – The *information view* provides the information engineering perspective of business solution architectures. The view is concerned with requirements for information resources. This will typically include a definition of what information will be stored, and what business rules this should adhere to.

Work view – The *work view* is concerned with the *how* of the business. This view is expressed in terms of work activities, associated resources, work locations, and needed information. In our context, an important goal for defining a work view is to determine the most effective ways in which the work activities can be supported by IT solutions. An important aspect of the work view is therefore the distinction between manual work and automated work.

Application view – An *application view* describes the business realisation activities that will be automated. It defines how the automated parts of the work view will operate, which information resources are needed, and how technology will be used to achieve this. The application view is positioned in the centre of Figure 4.12 to emphasise the forces that bring about changes to this view. The dominant forces that will change the application view come from the business and the technology sides. Any changes to those views will directly or indirectly result in changes in the application view.

Technology view – The *technology view* focuses on the technology needed to *facilitate* the other components of the architecture. While the business view focuses on *what* an organisation does, this view focuses on *what with*.

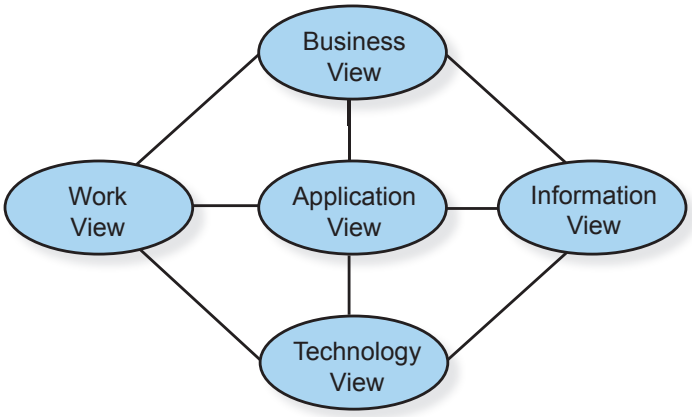

Fig. 4.12 Five views on an enterprise and its IT

Each of the five views from Figure 4.12 defines a specific (basic) subject about the enterprise one is interested in.

4.4.2 The Integrated Architecture Framework.

The Integrated Architecture Framework [30, 45] was developed by Capgemini as a means to structure architecture projects. It has evolved out of several decades of

experience in the field of architecting. The diagram in Figure 4.13 shows the basic structure of Integrated Architecture Framework. The framework is broken down into *aspect areas* (business, information, information systems and technology infrastructure) and *abstraction levels* (contextual, conceptual, logical and physical). To address disciplines of Security and Governance, the framework also recognizes two distinct views dealing with these issues.

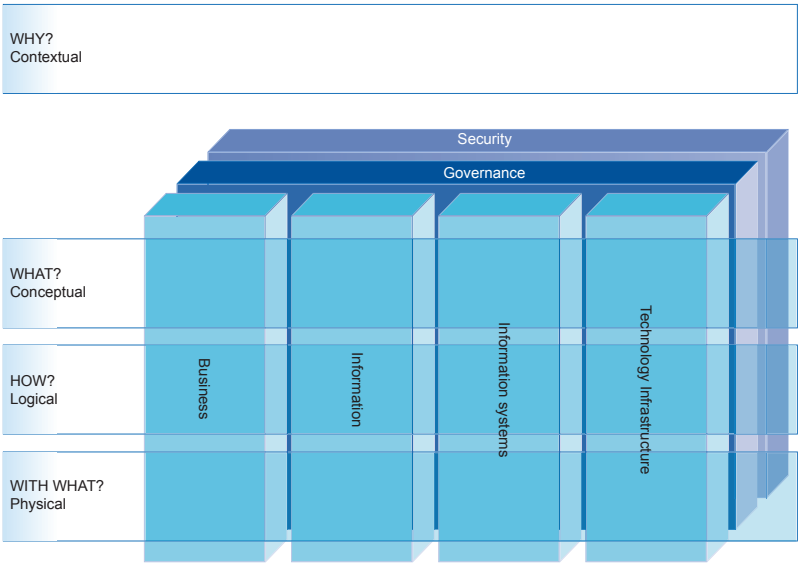

Fig. 4.13 The Integrated Architecture Framework

Abstraction allows a consistent level of definition and understanding to be achieved in each area of the architecture. The Integrated Architecture Framework defines four levels of abstraction:

Contextual level – The contextual level is characterised by "*Why?*" It is not about understanding what the new architecture will be; the level helps to identify boundaries (i.e. scope and objectives) for the new architecture and its context. Specifically, this level focuses on the business aspirations and drivers, capturing the principles on which the architecture will be based.

Conceptual level – The conceptual level is characterised by "*What?*". The requirements and objectives are analyzed and elaborated, ensuring that all aspects of the scope are explored, that relevant issues identified, and resolved, without concern about the way in which the architecture will be realised.

Logical level – The logical level is characterised by "*How?*". The level helps to find an ideal solution that is independent from implementation. From this, several "solution alternatives" can be developed that either provide the same outcome, or alternatively "test" different priorities and scenarios to understand the impli-

cation of different potential outcomes. The outcome of logical level analysis is the vision of the desired to-be state.

Physical level – The physical level is characterised by "*With what?*". It is about determining the real world structure and organisation, and is concerned with translating the logical level 'desired' structure and organisation into an implementation-specific structure, bounded by standards, specifications and guidelines. At the physical level, the outcome is a description of how the desired state will be achieved. The physical level provides standards, guidance and a degree of specifications within which further design will take place.

The Integrated Architecture Framework recognizes four "Aspect Areas", which focus exclusively on the core aspects of the overall architecture:

Business aspect – The business aspect area adds knowledge about business objectives, activities, and organisational structure.

Information aspect – The information aspect area adds knowledge about information that the business uses, its structure and relationships.

Information systems aspect – The information system aspect area adds knowledge about types of information systems (packaged or bespoke) that can automate and support the processing of information used by the business.

Technology infrastructure aspect – The area of technology infrastructure aspect adds knowledge about types and structure of components that support the information systems and actors. These may be hardware or network related. They may include fundamental services such as databases, etc. and key security and other commodity shared services.

The security and governance views respectively focus on:

Governance view – The governance view focuses on manageability and quality of the architecture implementation, required to satisfy the service levels (SLAs) required by the business for its processes and systems. The artefacts for this area are all fundamentally defined within the core aspects areas although the outcome from this aspect area will be new specialized Services and Components to deliver the governance.

Security view – The security view focuses on knowledge to mitigate known risks to the architecture implementation. The artefacts for this aspect area are all fundamentally defined within the core aspects areas. The outcome from this aspect area will be new specialized Services and Components to deliver the required security.

4.4.3 The ArchiMate framework

The ArchiMate project also produced an architecture framework. This framework is depicted in Figure 4.14. The framework identifies a *business*, *application* and *technology* layer, as well as three columns dealing with *passive structure*, *behaviour*

and *active structure.* Even though the distinction between the business, application and technology layers are an integral part of the ArchiMate standard language, the language has been defined in such a way that in principle an arbitrary number of abstraction layers can be stacked which are inter-linked with services, where each layer has the same core concepts as shown in Figure 4.15.

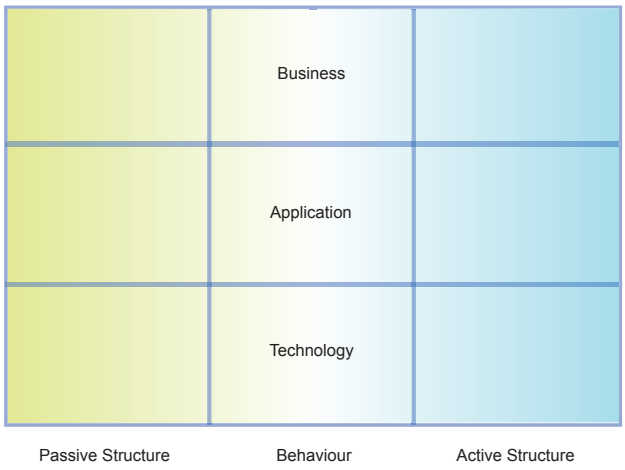

Fig. 4.14 The ArchiMate framework

Figure 4.15 also illustrates the elegance of the ArchiMate language. Rather than offering a plethora of distinct modelling constructs, the language comprises five core modelling concepts (object, service, behaviour element, interface and structure element), which re-appear at each of the layers. In the complete modelling language one will indeed see these five concepts repeated at the three identified levels. However, in essence, they remain the same core concepts. This can be likened to UML's Superstructure [92], where more specific modelling concepts are specialisations from a generic set of basic concepts.

4.4.4 The Zachman framework

The English language, as well as most other languages, contains a class of words called the interrogatives [137]: *Which*, *when*, *how*, *what*, *why*, *where*, *whose*, etcetera. These words may be used to formulate questions concerning situations, people, or any other phenomenon we may perceive or conceive. In other words, we may use these interrogatives to identify different relevant aspects of an enterprise. By using questions based on the interrogative words, insight may be gained into different aspects of an enterprise, such as: *Actors*, *timing*, *processes*, *functionality*, *rationale*, *purpose*, *locality*, *structure*, *ownership*, etcetera. These aspects form the core of the

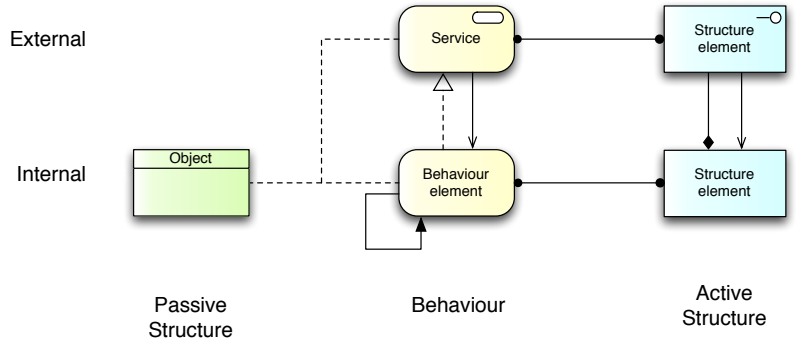

Fig. 4.15 Core concepts of the ArchiMate language

Zachman enterprise architecture framework [155] as depicted in Figure 4.16. The columns cover the following aspects (compiled/taken from [128, 155]):

Data (*what*) – This column addresses data needed for the enterprise to operates, the structure of the data, the way it will be stored, etc.

Function (*how*) – This column is concerned with the operation of the enterprise. It translates the mission of the enterprise into successively more detailed definitions of its operations.

Network (*where*) – This column is concerned with the geographical distribution of the enterprise's activities.

People (*who*) – In this column, one is interested in the people who do the work, the allocation of work, and people-to-people relationships within the enterprise.

Time (*when*) – Time is abstracted out of the real world to design the event-to-event relationships that establish the performance criteria and quantitative levels for enterprise resources.

Motivation (*why*) – The why column is comprises the descriptive representations that depict the motivation of the enterprise, and will typically focus on end-means-end, where ends are objectives (or goals) and means are strategies (or methods).

In addition to the six columns, the Zachman framework identifies six layers (compiled/taken from [128, 155]):

Scope – The first architectural sketch is a "bubble chart", which depicts in gross terms the size, shape, spatial relationships, and basic purpose of the final structure. It corresponds to an executive summary for a planner or investor who wants an estimate of the scope of the system, what it would cost, and how it would perform.

Enterprise or business model – Next are the architect's drawings that depict the final building from the perspective of the owner, who will have to live with it in the daily routines of business. They correspond to the enterprise (business) model, which constitutes the design of the business and shows the business entities and processes and how they interact.

System model – The architect's plans are the translation of the drawings into detailed specifications from the designer's perspective. They correspond to the system model designed by a systems analyst who must determine the data elements and functions that represent business entities and processes.

Technology model – The contractor must redraw the architect's plans to represent the builder's perspective, which must consider the constraints of tools, technology, and materials. The builder's plans correspond to the technology model, which must adapt the information system model to the details of the programming languages, I/O devices, or other technology.

Detailed representations – These correspond to the detailed specifications that are given to programmers who code individual modules without being concerned with the overall context or structure, and/or process designers who design detailed workflows.

Functioning enterprise – Finally, a system is implemented and made part of an organisation. This is a view of the program listings, database specifications, networks, and so forth that constitute a particular system. These are all expressed in terms of particular languages.

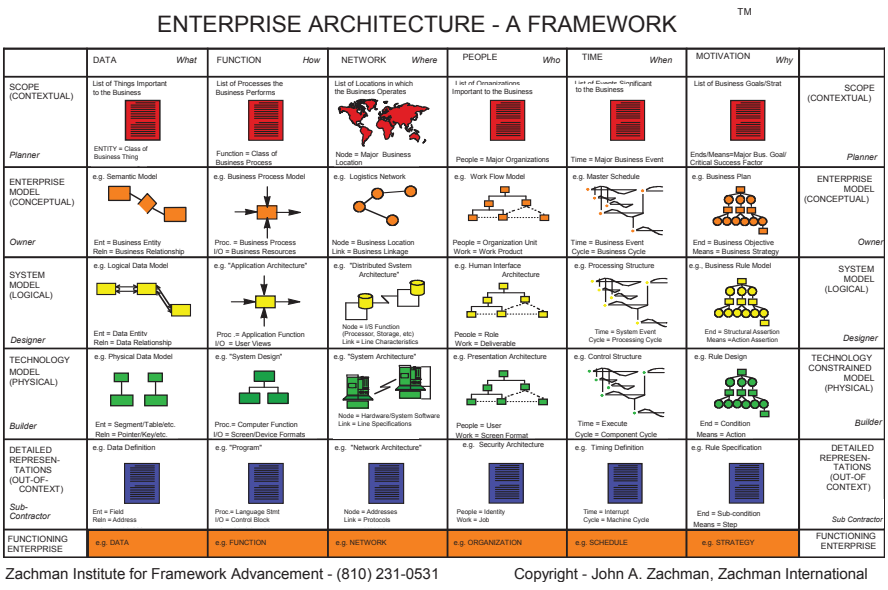

Fig. 4.16 The Zachman Framework

4.4.5 The Open Group's Architecture Framework

TOGAF [139], The Open Group's Architecture Framework, is organised into three Sections: *Architecture Development Method*, *Enterprise Continuum* and *Resource Based*. This is illustrated in Figure 4.17. Each of the Sections provides some guidance on what the outputs of a TOGAF-derived architecture should be and how they should be structured. Below we provide a discussion of these Sections. This discussion is taken from [136].

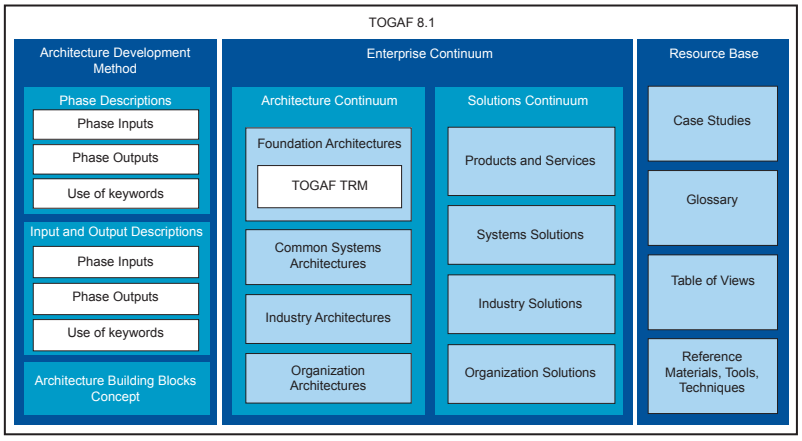

Fig. 4.17 TOGAF 8.1 content overview

4.4.5.1 The TOGAF Architecture Development Method

The *Architecture Development Method* (ADM) explains how to derive an organisation specific enterprise architecture that addresses business requirements. ADM is a major component of TOGAF and provides guidance for architects on a number of levels:

- It provides a number of architecture development phases (e.g., Business Architecture, Information Systems Architectures, Technology Architecture) in a cycle, as an overall process template for architecture development activity;
- It provides a narrative of each architecture phase, describing the phase in terms of objectives, approach, inputs, steps, and outputs. The inputs and outputs Sections provide an informal definition of the architecture content structure and deliverables;
- It provides cross-phase summaries covering requirements management, phase input, and phase output descriptions.

The ADM is primarily a process framework. As such, it can be used in conjunction with different product frameworks, such as the IAF, ArchiMate and Zachman frameworks.

4.4.5.2 The Enterprise Continuum

The *Enterprise Continuum* provides a model for structuring an architecture repository – a "virtual repository" of all the architecture assets. This is based on architectures and solutions (models, patterns, architecture descriptions, etc.) that exist both within the enterprise and in the IT industry at large, and which the enterprise has collected for use in the development of architectures. Architecture Building Blocks reside within the Enterprise Continuum. At relevant places throughout the TOGAF ADM, there are reminders to consider which architecture assets the architect should use.

4.4.5.3 The TOGAF Resource Base

The TOGAF *Resource Base*, "the reference content", is a set of resources, guidelines, templates, background information, etc. provided to be of assistance to the architect in the use of the ADM.

4.5 Dimensions for architecture frameworks

In the previous Section we discussed four architecture frameworks. There are, however, many more frameworks in existence. In addition to Tapscott and Caston, Zachman, IAF, TOGAF, and ArchiMate, some other publicly available frameworks are: the CRIS framework [91], Multiview [153], Kruchten's $4+1$ framework [77], RM-ODP [64], eTOM's business process framework [138], GERAM [61], IAF [30, 45], EEF [98] and DYA [148]. In addition, several organisations use their own internally created architecture framework. Usually, these frameworks have been constructed by their respective authors in an attempt to cover all relevant aspects of the design/architecture of some class of systems/enterprises.

Each of these frameworks covers several dimensions and comprises a number of cells. In the remainder of this Section we will explore several of these dimensions. The list of dimensions is based on work reported in several other publications [42, 49, 56, 64, 67, 77, 78, 91, 101, 104, 105, 107, 135, 139, 138, 148, 153, 154].

Most notably, the authors of [49, 56, 67, 154] have endeavoured to distinguish the dimensions used to span architecture frameworks. Below we provide a synthesis of these latter two works combined with our own observations from the other frameworks in existence. In doing so, we will distinguish three classes of dimensions:

1. Dimensions pertaining to the *subject* of a view.

2. Dimensions dealing with the *purpose* of views.
3. Dimensions concerned with the *form* of views.

When comparing these dimensions in architecture frameworks with the well-known notion of *scope* as used in project/program management, we will see that this notion of scope is potentially related to (selections made in) each of these (classes of) dimensions. Hence the lists provided below can be used to scope the assignment of projects/programs.

4.5.1 Subject dimensions

We now first turn our attention to those dimensions for enterprise (engineering or architecting) frameworks concerned with the subjects of the cells in the framework. Our aim is not to provide an exhaustive list of dimensions, nor do we claim that the list is orthogonal. We merely aim to provide an overview of some of the dimensions used.

Range – The range of the domain that is under consideration. Example classifiers are: single business processes, business units, the entire enterprise, and the entire value chain or ecology.

Construction abstraction – The level of abstraction from the actual construction of the enterprise. As used in the pizzeria example, example classifiers would be: valuation (*what value does it provide to its environment/ecology?*), function (*which functions does it provide in creating this value?*), construction (*how does it realize these functions*?).

Implementation abstraction – The level of abstraction from underlying technologies (including IT, human technology, machines, etc). Example classifiers (inspired by [63]) would be: conceptual (*what is needed?*), logical (*how will it be constructed/composed?*) and physical (*with which technologies and infrastructural elements will it be implemented?*). Note: the construction may use functions provided by other parties within the environment/ecology.

Enterprise system types – When considering an enterprise, several system types can be discerned covering different facets of the enterprise. Some example system-types are: business system, information system, production system, IT infrastructure and management & control system. Most architecture frameworks, in line with their IT roots, identify a dimension based on system types, which focuses on *business realisation through IT*. In these latter frameworks we usually find classifiers (system types) such as: business, information systems, applications and infrastructure.

Aspects of dynamic systems – Enterprises are dynamic systems. In such systems there will be actors/agents which exhibit behaviour, which will impact on objects in the domain. Typical classifiers are therefore: behaviour (*what happens?*), passive structure (*what is it happening to?*), and active structure (*what is doing it?*).

System qualities – Models may focus on different *qualities* of systems. Several quality attributes are in existence [65, 66]. Some examples are: efficiency, security, functionality and reliability.

Interrogatives – The English language, as well as most other languages, contains a class of words called the *interrogatives* [137]. These lead to a natural set of classifiers: which, when, how, what, why, where, whose, ...

4.5.2 Purpose dimensions

With the purpose of an enterprise architecture results, we refer to the combination of the communicative goal of the result, the intended audience and the process context:

Goal of the result – In the introduction of this Chapter already identified four types of goals for which architecture results may be created: specifying, deciding, informing and contracting.

Audience – The audience of the result. Some examples are: decision makers, actors in the transformation (architects, designers and engineers), actors in the current/future enterprise.

Transformation stage – The focus with regard to the stages in an enterprise's development. Some possible classifiers might be: existing situation (as-is), requirements on the future enterprise (as-desired), and its design (to-be).

Planning horizon – The planning horizon with which we regard the (future) enterprise. Typical examples are: current, near-term and long-term.

4.5.3 Form dimensions

As discussed before, the form of an enterprise architecture result refers to the communication style and languages used. One may, for example, opt for an intangible form or a tangible form such as a graphical model/view, a textual model/view, a combination of the latter, or even an animation. The actual form of a result should be in line with its intended purpose and audience.

In the case of intangible results, three dimensions (related to *pragmatic quality* of models [24]) can be applied:

Level of understanding – The level of shared understanding concerning the architecture results.

Level of agreement – The level of shared agreement with regards to the architecture results.

Level of commitment – The level of commitment concerning the architecture results.

The distinction between a level of agreement and a level of commitment may sound artificial. Nevertheless, in practice there is a distinction to be made between

agreeing that something is right for the enterprise, and indeed accepting the consequences these choices will/may have on a stakeholder's goals and concerns. This becomes even more critical when such consequences materialize after the decisions have been made, since the transformation process of an enterprise is likely to impede on the concerns of many stakeholders. An important aspect in the process of enterprise architecting is therefore the creation of a shared conceptualisation of the direction, which the enterprise transformation process should take. This makes it important to obtain the right levels of understanding, agreement and commitment at the right time. The enterprise architecture ideally is the embodiment of this shared conceptualisation.

With regards to tangible results, we may distinguish the following dimensions:

Level of detail – The level of detail to be covered in a result, in other words how detailed the results mimic the intended/existing enterprise. In [78] three levels are suggested: detailed, coherence and overview. Views at a detailed level typically focus on one cell of an architecture framework, while views at a coherence level will generally focus on the relationships between the cells within one dimension. Views at the overview level will aim to provide an overview covering multiple dimensions at the same time.

Level of precision – The precision at which the results are specified. A possible way to express the level of precision would be in terms of its level of formality, referring to the level at which it would allow for mathematical/automated interpretation and/or manipulation. Some example levels would be [101]: informal, semi-formal and formal. Informal would typically be a graphical sketch or a loose narrative description. Semi-formal would be using a controlled (graphical or textual) language, i.e. limiting the allowed syntactic variation, yet still without a well-defined semantics. Formal then implies the use of a (restricted) language with a well-defined semantics, enabling a precise and unambiguous interpretation of the results.

In colloquial use, the levels of detail and precision tend to be confused. When representing something at an overview level, one tends to misinterpret this as an excuse to provide a vague and imprecise description.

4.6 A methodical perspective on the creation of results

Enterprise architecture results will typically feature numerous models and views. In an architecture context, these models are created in line with so-called viewpoints [60]:*A specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.* To some extent a viewpoint establishes a *method* for the creation of views, in particular where it concerns "*the techniques for its creation and analysis"*. In [122, 152] a framework was proposed to dissect a modelling method into a number of aspects:

Way of thinking – Articulates the assumptions on the kinds of problem domains, solutions and modellers. This notion is also referred to as *die Weltanschauung* [126, 153], *underlying perspective* [83] or *philosophy* [12].

Way of modelling – The way of modelling provides an abstract description of the underlying modelling concepts together with their interrelationships and properties. It structures the models, which can be used in the information system development, i.e. it provides a language in which to express the models.

Way of working – Structures (parts of) the way in which a system is developed. It defines the possible tasks, including sub-tasks, and ordering of tasks, to be performed as part of the development process. It furthermore provides guidelines and suggestions (heuristics) on how these tasks should be performed.

Way of managing – The managerial aspects of system development. Originally this was referred to as the *way of controlling*. It includes such aspects as human resource management, quality and progress control, and evaluation of plans, i.e. overall project management and governance (see [72, 127]).

Way of supporting – The support to system development that is offered by (possibly automated) tools. In general, a way of supporting is supplied in the form of some computerized tool.

The resulting framework is shown in Figure 4.18. As synonyms, one may refer to a way of working as a *(modelling) approach* and to a way of modelling as a *(modelling) technique*. The *way of thinking* and *way of modelling* of a method are strongly tied to the subject dimensions in architecture frameworks since these dimensions determine *what* is to be modelled. The *way of modelling* is also tied to the form dimensions, in particular the levels of detail, specificity and formality needed in the results.

When using an architecture framework, a large number of viewpoints can be associated. Starting with the subject dimensions:

- For each cell in the framework there is likely to be a specific viewpoint. For example: the business aspect at the conceptual level, or the technology aspect at the logical level.
- For some complete dimensions, there will be viewpoints covering the entire dimension. For example, the entire conceptual level.
- There are likely to be viewpoints focusing on the relationships between a cell or a dimension, and its direct neighbours. For example, relating the conceptual level to the logical level, or the business aspects to the informational aspects.

In addition, mainly fuelled by the purpose, there are likely to be many, many, ad-hoc viewpoints depending on specific concerns, audiences and purposes. For example, a view showing a design alternative at the physical level, with an associated cost/benefit analysis at the conceptual level, aiming at answering the concern of the CFO.

In creating/selecting an architecture framework, organisations should focus on a framework that is based on structural considerations based on the stakeholder's concerns that are key to the enterprise and require enduring attention. Ad-hoc viewpoints (and ensuing views) can always be added when needed. This most likely

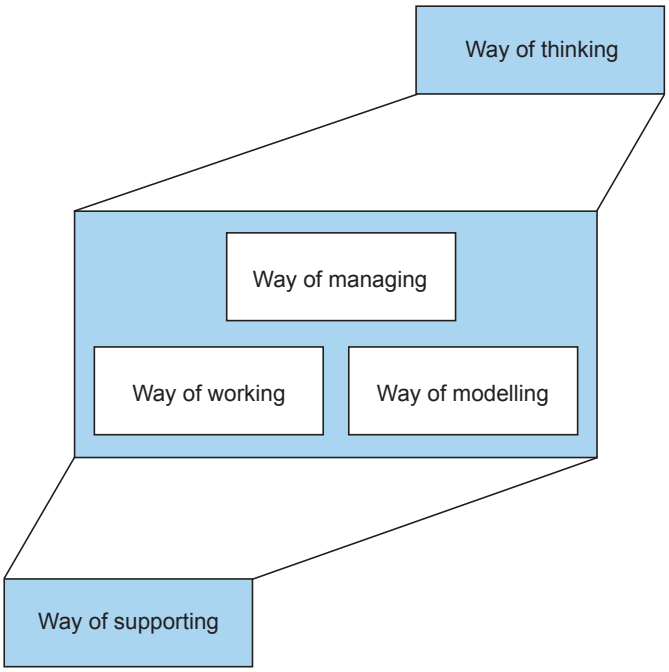

Fig. 4.18 Aspects of a modelling method

implies that the framework will primarily comprise subject dimensions, focusing on view(point)s providing stakeholders with insight, as well as steering abilities, relevant to their concerns. Additional ad-hoc view(point)s can be added on top of that tuning the structural view(point)s to the specific needs/uses at hand.

4.7 The call for a unified notation

As mentioned in the pizzeria example, there is no single modelling language in which to model all relevant aspects of the pizzeria domain with a unified look-and-feel. In the pizzeria example, we showed several aspects of this organisation using modelling techniques such as: e3Value, DEMO and ArchiMate. Even though it is doubtful whether it is possible to have a single unified language covering all aspects of an enterprise, a well integrated language with a unified look and feel covering at least the core valuation, function and construction aspects of an enterprise is desirable [78]. Note that in expressing this desire, we refer to the *modelling* language used to create *models* of different aspects of an enterprise. When creating *views,* one is likely to use different styles and languages that are better attuned to the needs of the intended audiences. The models will be used primarily by architects and engineers to express the design of the enterprise in all of its richness.

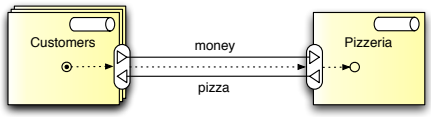

Fig. 4.19 Pizzeria at the valuation level

ArchiMate was designed to play the role of such a unified language. Even though we only use the ArchiMate notation for the infrastructural aspects of the pizzeria, it is indeed suitable to represent the construction diagrams used in the pizzeria case. Even more, the ArchiMate language allows for the definition of extensions, where new concepts are introduced as specialisation of existing ones and/or combinations of existing ones. Even though the concept of transaction, as used in our Pizzeria example, is not explicitly built-in the ArchiMate language, it can be constructed by defining a transaction concept being a collaboration between the two roles participating in the transaction, involving a number of processes (covering the request, promise, execute, state and accept phases) and associated orchestration. The same applies to the value exchange between the customer and the pizzeria, which can essentially be constructed in terms of ArchiMate's value and service concepts. When using such constructs, one would be able to produce diagrams such as shown in, Figure 4.19, Figure 4.20, Figure 4.21, Figure 4.22 and Figure 4.23, depicting an ArchiMate-ish version of Figure 4.1, Figure 4.2, Figure 4.3, Figure 4.4 and Figure 4.5 respectively. Note that Figure 4.21 and Figure 4.23 also illustrate how a slightly different perspective leads to additional insight. The original views took the *roles* as leading, by showing how the roles participate in transactions, while the flow of the transactions meanders over them. In the new views we took the *transactions* as leading, and show how the transactions are executed by the different roles.

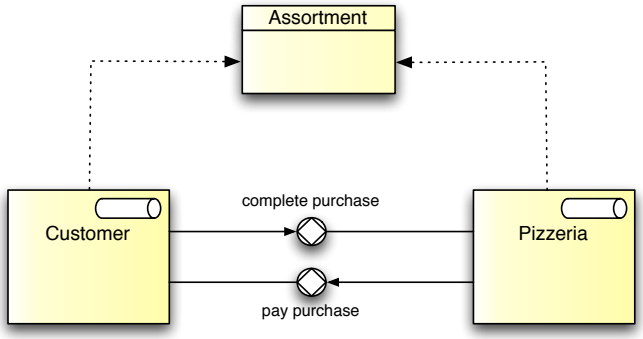

Fig. 4.20 Transactional view of value exchange

Figure 4.19 shows the value exchange between the customer and the pizzeria as a value exchange between business roles. Figure 4.20 shows the realisation of this

value exchange in terms of the transactions *complete purchase* and *pay purchase*. Figure 4.21 depicts the orchestration of the complete purchase and pay purchase transactions as being a collaboration between two business roles: *customer* and *pizzeria*. The construction of the pizzeria in terms of a completer, baker, transporter and their mutual transactions is shown in Figure 4.22. Finally, Figure 4.23 shows the orchestration of the transactions when the actual construction of the pizzeria in terms of the completer, baker and transporter is considered.

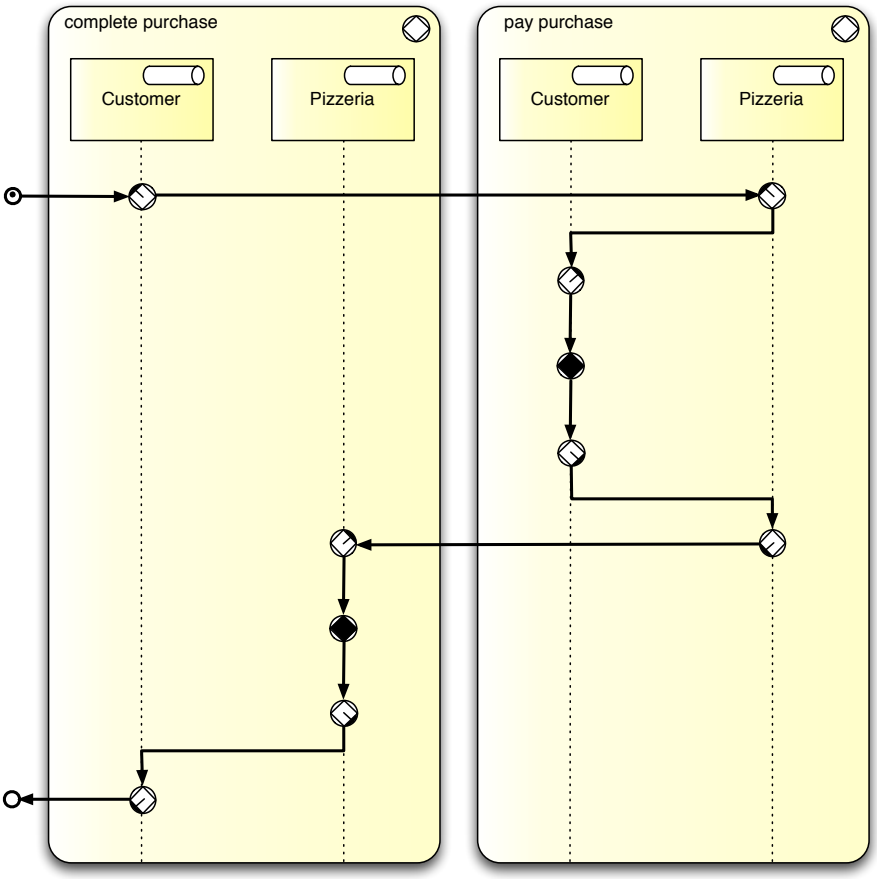

Fig. 4.21 Orchestration of transactions

In the further evolution, and possible integration, of modelling languages such as ArchiMate, DEMO and e3Value, there are some wise lessons to be learned from the development of a language such as YAWL (Yet Another Workflow Language [4]). In the development of this workflow language, the authors started by surveying patterns used in workflow specifications. In doing so, they first gained insight into the

constructions, which the language should be able to express naturally. For example, the desire to discuss/express the impact which a business principle such as "*Outside own organisation: Get money first!*" will have on the orchestration of the transactions between a customer and the pizzeria, requires a modelling language supporting an explicit notion of a transaction and the phases (request, promise, execute, state, accept) within a transaction. The same applies concerning the modelling of value exchanges between the pizzeria and the consumer. The desire to be able to reflect on the value being exchanged between the pizzeria and its consumers, or between any pair of business roles for that matter, requires explicit modelling constructs.

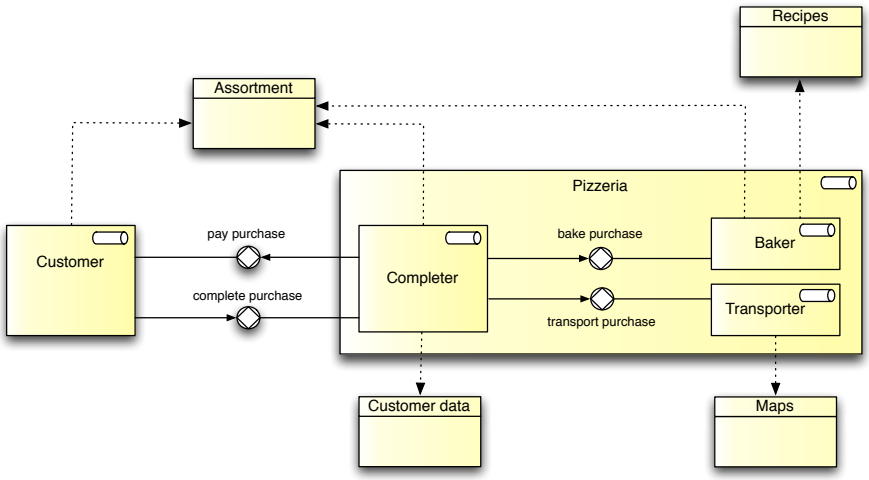

Fig. 4.22 Decomposition of the Pizzeria

When the ArchiMate language was developed initially, it was also based on a survey of needs voiced by the industrial partners in the project [69, 70]. In the meantime, however, insight into what one wants to do with enterprise architectures has evolved, and as a result also the demands on the modelling language used have evolved. In anticipation of such evolution, the ArchiMate language was equipped with an extension mechanism to indeed cater for such evolutions.

4.8 Summary

In this Chapter we have explored the results that may be yielded from enterprise architecting efforts. We have used the pizzeria example to exemplify some of these results, covering final deliverables, intermediary results and intangible results. Tangible deliverables/results include models, principles and views. We then moved on

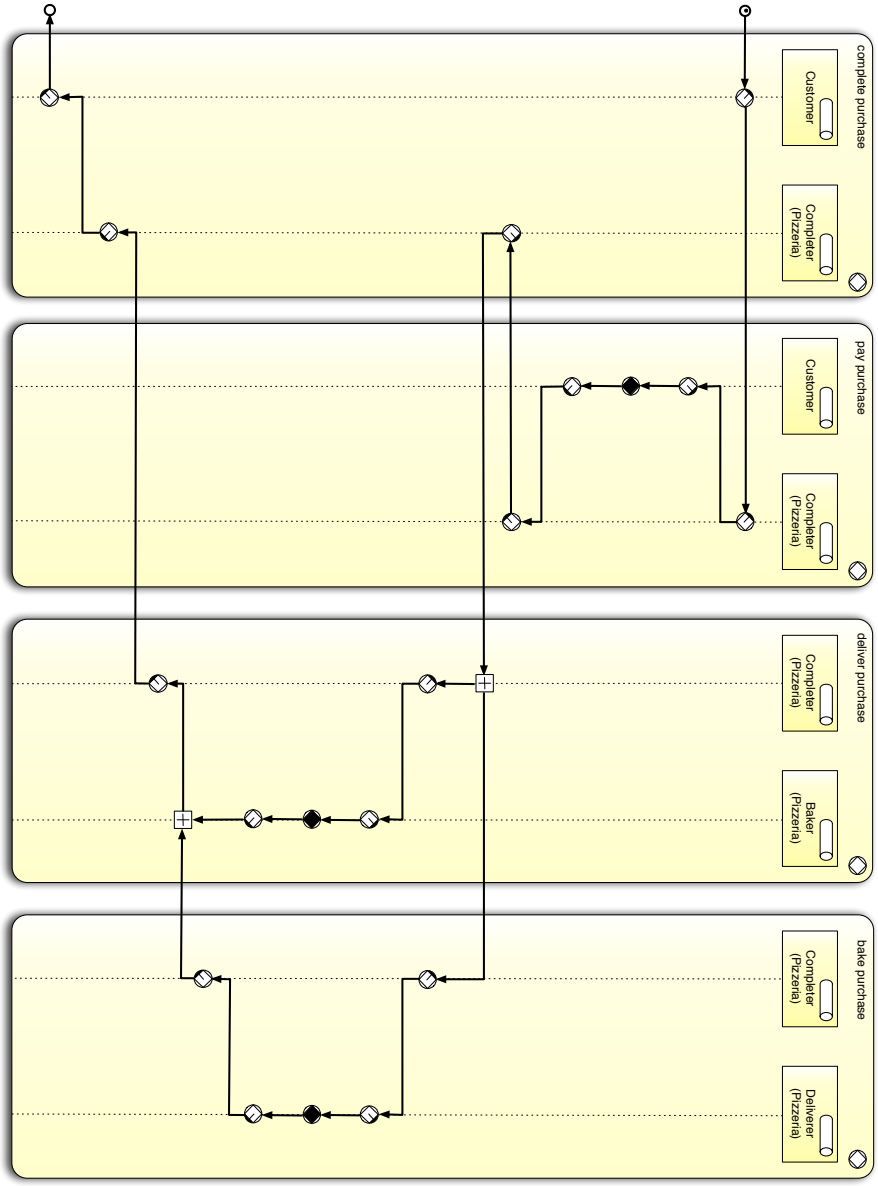

Fig. 4.23 Composed transaction orchestration

to the issue of quality of the possible results. Depending on the intended use, different quality requirements should be met. We identified four key types of usage of results: specifying, deciding, informing, and contracting, also covering justification of decisions made. This intended use has a great impact of the scope of the architecture results needed. Based on the usage requirements of the involved stakeholders and their concerns, the intended result can be designed in terms of its subject and form.

From our desire to have order and completeness in architecture results, we then turned our attention to architecture frameworks, and started by discussing some examples of the architecture frameworks in existence. As means to order architecture results and guard their completeness, a framework gives insight into the interrelationships of architecture results, and therefore enables the traceability of decisions and their impact.

To better understand the concepts underlying the multitude of frameworks, we surveyed some possible dimensions for frameworks, distinguishing between dimensions dealing with the subject, purpose and form. We also related the concept of viewpoint to a pre-existing framework for modelling methods distinguishing between a way of thinking, a way of managing, a way of supporting and a way of modelling. A strong link exists between the subject dimensions of an architecture framework and the way of thinking and way of modelling of the modelling methods used in creating the results, as well as between the form dimensions and the way of modelling of these methods. Even more, we took the position that an enterprise architecture framework should primarily focus on subject dimensions. We also identified an opportunity to define *scoping* as a project/program management notion in terms of the selection of different (classes of) dimensions.

Finally, we argued the case for modelling techniques for (the design-oriented perspective of) enterprise architectures that cover different aspects of an enterprise, while still offering a unified look and feel.

4.9 Discussion statements

1. Views that properly address the concerns of stakeholders are of more value than the underlying architecture as a whole.
2. Artist impressions are more effective than detailed models.
3. An enterprise should standardise on one architecture framework and stick to it.
4. When copying the enterprise architecture of a competitor, one also imports its underlying vision and strategy. In other words, if an enterprise wants to have a unique competitive edge, it needs its own architecture.

Chapter 5
The Process of Enterprise Architecting

5.1 Introduction

As we have seen in Chapter 4, the purpose for which a particular architecture is created largely determines the kinds of results needed. This purpose also impacts on the form and execution of a good architecture *process*. However, the purpose is not the only major influencer. Another major influencer is the *multitude* of stakeholders, which over time need to agree on the direction of the enterprise transformation. In this change process the insights gradually evolve, while the decisions taken during the process may change the direction of this process. Therefore, the architecture process can also not be a linear one. It strongly depends on situational factors, and therefore cannot be served by a one-size-fits-all approach.

The core process of enterprise architecture encompasses creating, applying and maintaining the architecture for its intended purposes. In the way this process is executed, we see best practice patterns and styles emerge, sometimes even materialising in terms of "architecture schools". Whether a specific style or pattern is effective or not, not only depends on the purpose of the architecture, but also on the architecture maturity of an organisation, it's management style and culture. Process quality criteria, mainly derived from purpose and maturity, are required to enable a choice for an effective and efficient architecture process. The maturity, the situational criteria and preferred architecture patterns will all influence how to organise the continuous improvement of the architecture function.

The remainder of this Chapter is structured as follows. We start by describing the core elements – *create, apply and maintain* – of the process of enterprise architecting. Next, we will give examples of patterns, as well as best practices, to execute this core process. Subsequently, we will reflect on the importance of architecture maturity, introducing an architecture maturity model. We then continue by showing how the core architecture process can be organised to implement the *plan – act – learn* cycle of the enterprise as a whole, taking outsourcing of architecture roles as an option. Finally, we summarise the architecture process concepts introduced and highlight their coherence.

5.2 The core process of enterprise architecting

Enterprise architecting involves a number of core processes. First of all an enterprise architecture has to be *created*. To be really useful as an instrument within an enterprise, the architecture has to be *applied* in line with its purpose. To remain useful in a changing world, an enterprise architecture should be *maintained* as well. Each part of the architecture process – *create, apply and maintain* – will to a large extent be influenced by the purpose of a specific architecture.

5.2.1 Creating enterprise architecture

Let us first elaborate on the act of creating an enterprise architecture. As we have seen in Chapter 4, the purpose of enterprise architecture will determine the results which should be produced. In general, both tangible and intangible results will be strived for, each requiring its own process. What does such a process look like?

Consider the following sketch of an enterprise architecting process, mimicking a basic project setup:

> *Inspired by Chapter 4, one would start by selecting the (tangible and intangible) results needed. Then one would carry on by defining the activities needed to produce these results and order them in a plan. Subsequently, one would have to arrange staffing of the project and request formal permission to initiate the project. Finally, one would disappear out of sight, and re-appear about a half year later with beautiful principles, models and views.*

This is a rather naïve approach, which is deemed to be totally ineffective. To indeed realise the intention of enterprise architecture – enabling the steering of change – much more is needed. For instance, a shared conceptualisation should emerge among stakeholders about:

- a **to-be** situation,
- the **as-is** situation,
- any constraints that should be met,
- purposes of the enterprise architecture that are met and those that are as yet unmet.

Even more, the concept of enterprise architecture might be a new means to the organisation, in which case an introduction is called for. In the organisation several large cultural differences may exist as well, e.g. between *thinkers* and *do-ers*, between *domain experts* and *management*, between *sales* and *operations*, et cetera. This all opens up new requirements about how to communicate about, and arrive at, enterprise architecture results, about the benefits of (an) enterprise architecture and about the process which is followed in creating enterprise architecture. For this kind of change you have to involve, get access to and get buy-in from important stakeholders before, during and after the “enterprise architecture project”. Communication is an important aspect in this.

This raises the question: *Which stakeholders are required, how and when should they be involved, and how (and to what extent) does this depend on the purpose of the architecture and context in which it will be created*? Best practices learn that it is important to deal with the multiplicity of stakeholders and their concerns at the right time. Be aware that stakeholder needs and expectations may change over time. Even more, the set of stakeholders involved may change over time as well, more or less necessitating a (time-boxed) iterative approach. We limit ourselves to a few examples:

- Suppose an enterprise architecture is mainly used for decision making on an intended business transformation. Then a **to-be** *description*, possibly in the form of an sketch, is probably more relevant than specific guidelines on how to arrive at the **to-be** situation. This in turn means that mainly managerial and context stakeholders need to be involved in a more brainstorming way of working, in order to achieve common understanding and commitment.
- Assume an enterprise architecture has to be created in a context where "*don't ask why but ask how high*" is the predominant attitude. Instead of consultative and consensus building workshops, the way of working here will include more directive and elaborating workshops.
- Envisage an organisation for which enterprise architecture is a new endeavour. Then the enterprise architecture team should consider to keep the more complex and abstract intermediate enterprise architecture results for internal use only and to communicate with all stakeholders in terms of the added value of applying enterprise architecture.
- The concern of the IT manager about which IT to procure, is valid and should be dealt with separate (generally: later) from formulating business strategy impact.
- The concerns of the security officer regarding the potential impact on privacy regulations, when storing customer data, should be included right from the beginning.

In shaping the creation process, of course more general project and change management criteria apply as well [52]. As discussed in Chapter 4, the purpose of the enterprise architecture determines the level of quality required from the deliverables.. Together with constraints in duration, net time or money, this in turn will determine the feasible level of detail and accuracy. In deciding which staff members should be involved, we should now explicitly include criteria not only about knowledge but also on position, influence and stakeholder network.

5.2.2 Applying / using enterprise architecture

We now turn our attention to the application and actual use of enterprise architecture. In Chapter 3 we discerned a regulation-, a design- and a pattern-oriented perspective on enterprise architecture. In Chapter 4 we elaborated this into four main goals of enterprise architecture: *deciding*, *specifying*, *informing* and *contracting*. In each

of those four main goals, a shared conceptualisation is needed on issues such as opportunities and problems, strategies, possible solution alternatives and their trade-offs, and finally the chosen solution alternative. Each of those goals of enterprise architecture has their specific needs for a process.

We start by looking at situations where enterprise architecture is used to support decision making. To be effective as a steering instrument, enterprise architecture should be embedded in the "ordinary" steering processes of an organisation, for which Figure 5.1 shows an example. In the phases of *drafting needs* and *business case*, we use architecture mainly for the purpose of shared conceptualisation in terms of principles and a high level design. This helps in scoping the ambition, overseeing complexity and risks, and finally deciding if the organisation actually wants to execute the depicted change, i.e. answering the question: "*do we want to, and are we able to, do this?*" Typically, the latter decision will be taken by senior management and key stakeholders. When a positive decision has been made, the next question is how to realise and control the intended change. In this phase, enterprise architecture helps to plan the change and ensure compliance with the principles. In trading off solution alternatives, it will appear that not all principles can be fully met in any of them. Take for example the principles "service oriented" and "proven solutions", then in selecting an alternative with a higher priority for proven solution, the enterprise architecture will help to (1) make explicit the consequence of "less service oriented" in the solution and (2) give additional guidelines necessary for contracting, e.g., prescribing service layering or legacy wrapping. All along the way, enterprise architecture will thus enable risk assessment and mitigation.

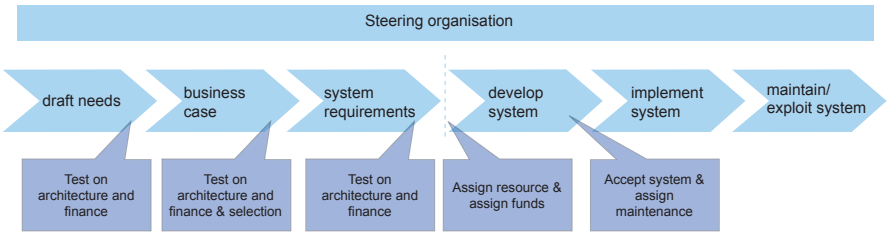

Fig. 5.1 Enterprise Architecture embedded in the steering process of an organisation

A specific use of enterprise architecture in decision making we find in the planning of change at several levels in the enterprise. Often enterprise architects collaborate with program managers in the long-range planning of transformations, e.g. in the context of a *design authority*. This results in a *plateau planning*, defining plateaus as a steady state of the business, and projects to arrive at these plateaus. Enterprise architecture will help making the plateau descriptions complete as far as content is concerned, at the same time defining project scopes and project dependencies, thus enabling estimations of time, money, risks and feasibility. In the "Perla del Nord" case as introduced in the previous Chapter, we actually focused on the *create* process and did not really cover the *apply* and *maintain* processes. In

this Section we will remedy this, also exemplifying these processes. In the "Perla del Nord" case, a possible plateau could be a situation in which the pizzeria is able to act in the B2B-market, including the requirements on B2B-contracting and credit risk checking. Another plateau could be the broadening of the product portfolio with e.g. lasagne. Senior management can now easier decide on ambition levels of plateaus, while program management can decide on how to arrange their program in terms of well scoped and feasible projects. In the example of "Perla del Nord", such projects could be "*marketing B2B service offering*", "*make available baking car/bus*", "*enable automated credit checking*", "*authorisation of B2B-employees*" and "*multi-channel offering for B2B*". The combined insights per intended program in turn can support the enterprise portfolio management in its investments in enterprise assets, aligning the initiatives, preventing overlaps and choosing the order of programmes. Summarising, the process of decision-making requires from enterprise architecture timely and stepwise growing insight in consequences of decisions. This insight to enables *informed* trade-offs in terms of time, money, risks and feasibility, i.e. laying the foundation for *informed governance*. This process of decision-making is quite diverse: in several phases, different stakeholders need to be involved, also depending on the outcome of trade-offs made in between.

Special attention is needed to make the link between enterprise architecture and programmatic steering of change (see Subsection 2.4.2) effective. This should already start at the level of *governance*. Enterprise architecture supports corporate governance by giving insight into substance and coherence of the entire value chain. At the same time, architecture itself needs to be embedded in the overall change and governance processes of the enterprise. *Projects* aim to realise parts of the to-be situation, in which they have to comply with the guidelines and the structure imposed by the enterprise architecture. To aid projects with this compliance, they should be accommodated with selections from the enterprises architecture, which are relevant to the scope of the project, additional viewpoints suitable to the concerns of the project's stakeholders, as well as operational criteria to ascertain the compliance of the project's result to the enterprise architecture. TOGAF's phase G [139] specifies this connection between the enterprise architecture and a specific project in a so-called *architecture contract*. Subsequently, the project has the responsibility (and freedom) to develop the architecture for their solution within the context of the architecture contract. Other sources use terms such as Project Start Architecture (PSA) to refer to the same connection [41, 148]. *Portfolio management* can use enterprise architecture as a common language to coherently define the programmes needed. Indeed, the enterprise architecture will show which intended components contribute in what extent to which goals and strategy, enabling underpinned choices in adding or removing parts of the organisation and technology. At the same time, enterprise architecture needs an overview of the programmes in the portfolio as an input. Finally, where *programme management* focuses on the managerial aspects of a body of projects, enterprise architecture ensures the cohesion between the product aspects of these projects.

We now turn our attention to the role which enterprise architecture can play in restricting and guiding design freedom, as required in specifying and contracting.

This role requires processes to formulate these restrictions and to ensure the compliance to them. In formulating the design restrictions, views are made for a part of a transformation, mainly indicating relevant principles and scoping in terms of sub-systems (component-systems and/or aspect-systems).

Suppose, "Perla del Nord" has decided to strive for the first plateau, i.e., to indeed enter the B2B-market. For the project "*make available baking car/bus*", such design restrictions could take the form:

- relevant principles:
  - bake while driving,
  - ability to handle growth in volume of a single delivery is more important than cost-efficiency,
- scoping of systems:
  - subsystem: the (combined) actor role of baker and deliverer, which we referred to as (functionary type) *transporter* in Chapter 4, including his cooperation with cook and order taker,
  - included aspects systems: vehicle (physical transporting & baking device) to support the transporter, including the information supply about the vehicle (location tracking, maintenance-guidelines for the driver),
  - excluded aspect systems: the recruitment / education / job descriptions for the transporter.

For the project "*multi-channel offering for B2B*", such restrictions could be:

- relevant principles:
  - all usual communication technologies for our customers should be supported,
  - re-use previously made models,
- scoping of systems:
  - subsystem: the actor role of the completer, as implemented in the functionary type *order taker*, including his cooperation with customer, cook and transporter,
  - included aspect systems: internet-channel & voice-channel, including educating the order taker in the use of that,
  - excluded aspect systems: recruitment / job descriptions for the completer.

Note that designers have freedom within the indicated restrictions, e.g., in the technology to be used for the voice channel. In communication with the marketing department of "Perla del Nord", it could well turn out that the *"usual communication technology*" for the target group is only instant messaging (IM) or Skype. This could mean that in "*implementing the internet channel*", e-mail will not be included.

In this last example, we see how a typical design decision emerges, which requires an iterative designer between designer and enterprise architect. Such a communication will result in four possible conclusions:

1. the enterprise architect changes (the project view on) the enterprise architecture, e.g. by including extra (new or already existing) principles or models, or by formulating better constraints;
2. the designer changes the intended design decision to adhere to the given design restrictions;
3. the enterprise architect decides to allow the decision made by the designer, at the same time adding this precedent to the enterprise architecture (as a kind of jurisprudence);
4. enterprise architect and designer agree that an intended design decision indeed has to be taken on project level, so that project and architecture align or at least do not contradict.

In the example of "Perla del Nord" this could mean that:

1. the enterprise architect informs the designer about already chosen but by accident not in the project-view included IM-standards, VoIP-standards and the data model for (retail) *client*,
2. the enterprise architect specifically emphasises the principle "we choose for open standards" and therefore the designer concludes that he will seek for a Skype-alternative,
3. since at this time no open standard based B2B-used alternative for Skype is available, the enterprise architect accepts and documents the proposed design decision to use Skype, and
4. the enterprise architect agrees that the intended decision not to implement customer-facing e-mail is compliant with the enterprise architecture indeed and can be taken at project level.

Since a guaranteed complete view for a project does not exist, this iteration helps the enterprise architect in finding a balance between over- and under-specifying. In order to ensure compliance, a compliance measurement process such as a formal audit or periodic review needs to be in place.

The created enterprise architecture is of interest to a wide variety of stakeholders. At the same time applying architecture is part of a change process, since this new means will impact the way of working for many. Therefore, on top of the communication before and during creating enterprise architecture (see Subsection 5.2.1), further communication is needed. To create additional buy-in, the communication should start at the level of awareness, to let stakeholders know that an enterprise architecture is in place and what benefits they generally can expect from it. Once awareness is in place, further communication can be targeted more specifically to the different stakeholders. For instance, decision makers need to understand the benefits, the typical insights they can expect from using it and the type of decisions it will facilitate. Project leads and lead designers need to understand the impact and value of expected restrictions at project level for the enterprise as a whole and the interaction with enterprise architects they can expect. As another example, the enterprise architecture can demonstrate its added value as a shared conceptualisation of the **to-be** situation and the transformation required. In the end, informing about enterprise architecture is analogous to normal communication: decide on target group and the

message per target group and subsequently derive the means to be employed (video, intranet, posters, road show, conference). Apart from this one-off communication, enterprise architecture also ensures the required continuous flow of information by giving and stopping access to enterprise architecture materials.

As we have seen in this Section, the different ways of applying enterprise architecture give a different emphasis in processes. For decision making the emphasis is on supporting, for ensuring compliance – e.g. in specifying and contracting – enterprise architecture will direct and in informing communication is paramount.

5.2.3 Maintaining enterprise architecture results

We now finally turn our attention to the act of maintaining enterprise architecture results. By maintaining the enterprise architecture we will understand both *monitoring* (business and/or technology) changes that might be relevant and *updating* the enterprise architecture. The effect of maintaining is that architecture results continuously and adequately reflect "reality" to a known and controlled extent. Now why and when is such maintenance of an enterprise architecture necessary? And how should it be done? Instead of giving a detailed description of the general ways for change management and impact analysis, we give some guidelines to adapt some common procedures. Sometimes, the creation of an enterprise architecture is a one-off event, e.g., when deciding not to go ahead with an intended change. The only relevant "maintaining" here would be to make this a conscious decision and enable traceability of this decision. Most of the time, however, such "creating of an enterprise architecture" is not a one-off event but rather an integral part of an overall (continuous) change process of an enterprise. Enterprise architecture results, in themselves, have a stable character and – because of the investments made in terms of involvement of stakeholders, time and money – represent a serious asset of the enterprise, deserving to be leveraged. A monitoring activity should therefore (regularly) assess whether the enterprise architecture should be updated by estimating the impact of typical change drivers. For instance, to what extent is the business changing, internally or externally: when should new legislation or supervisory guidelines be applied, when will innovations become available, are new goals pursued for, is a new strategy under development, are new products and services on their way? Or take the availability of people and means: will certain capabilities of people be less available, will capabilities become cheaper available elsewhere, is new technology or the phasing out of existing technology announced? Also internal reasons can be a change driver, such as the desire for a different and more efficient process, signals of issues/defects reported from projects or discussions with these projects. In the case of pizzeria "Perla del Nord", the project has chosen IM and Skype as technology for the voice channel, which raises several questions for EA such as "also use these channels for internal communication"; this means additional guidance is needed at the enterprise level, e.g. to prevent overlapping technology choices for IM. And of

course the stakeholders can change, by person or by role, which may be reason to see new concerns introduced and to give other concerns a different priority.

The monitoring task thus reveals the need to adapt the enterprise architecture. As a next step, the size and nature of the required adaptation has to be assessed: is this a minor change, to be solved by a well-localised update, or is this a major change leading to a new version, to be realised by a partial or complete (re-)creation of the enterprise architecture? A best practice rule of thumb is the amount of stakeholders involved: with one stakeholder it probably can be solved by an update. In the case of such an update, models and viewpoints will be improved and accentuated, as well as adding decisions to waive the enterprise architecture as a precedent to the documentation. Finally, the changes will be communicated to the relevant stakeholders.

When a major change is required, one should approach this as a (re)creation. In other words, all things stated before concerned the creation of an enterprise architecture applies: change management should start, the right stakeholders should be involved, etc.; a new architecture evolution cycle is started.

5.3 Patterns for enterprise architecting

For the core architecture processes we just introduced, several approaches with a proven track record have emerged. We will discuss their characteristics, reflecting on the assumptions under which such approaches can be successful. This will underpin our proposition that there is no one-size-fits-all approach to architecting, and that a situational approach is needed.

5.3.1 Architecture process patterns and architecture schools

We start with **TOGAF** [139], which contains the Architecture Development Method (ADM). ADM (see Figure 5.2) "*describes a method for developing an enterprise architecture*", in our terms for the architecture core process, including maintenance (called by TOGAF/ADM: phase H - Architecture Change Management). For each phase, ADM provides several guidelines and best practices. TOGAF/ADM does not prescribe any set of specific enterprise architecture deliverables; therefore it may be used in conjunction with the set of deliverables of another architecture framework. ADM describes a phasing model, starting from Framework and Principles via Architecture Vision (phase A) and Business Architecture (phase B) all the way to Migration Planning (F) and Implementation Governance (G). ADM points out that it can be applied iteratively in the whole process, between phases, and within a phase. ADM is continuously driven by the requirement process (the circle at the heart of Figure 5.2).

The Business-oriented Method for Information-planning (BMI) as described by [119] emphasises quality of models as a basis for long-term and high-quality

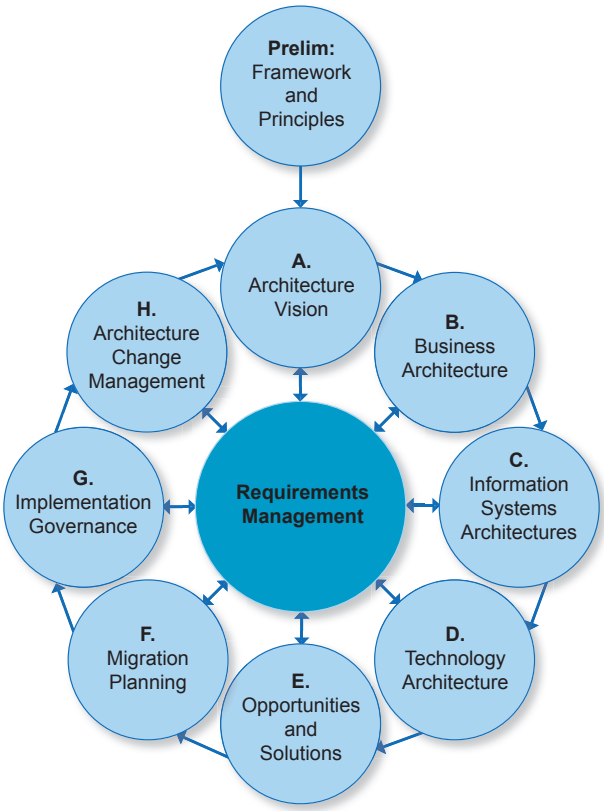

Fig. 5.2 Architecture Development Cycle

structure of the information supply. BMI has two central paradigms, namely (1) the distinction between essence and implementation of an organisation and (2) the distinction between business models and information models. This also determines the way of working: creating architecture begins with business models – preferably modelling the essence of the organisation first – and ends with the two types of information models and its interconnections. The way of working is focused on creating architecture and leaves much freedom for the way of interacting with the stakeholders.

The Design & Engineering Methodology for Organisations (DEMO) as described by [35] also emphasises quality of models to underpin high-quality decisions to direct both implementing the organisation and implementing information supply. Until recently, DEMO's way of working has been focused on shaping the actual architecture of organisations, leaving much freedom for the way of interacting with stakeholders. Recent research has tuned DEMO's way of working to specific needs, e.g. in general organisation design [86], in areas of organisation splitting – to enable Business Process Outsourcing, to use Shared Service Centres in the primary

business or to make (de)centralisation choices [81] – and in the area of application portfolio rationalisation [97].

DYA [148, 147] is an approach to enterprise architecture from Sogeti – Netherlands. DYA stands for DYnamic Architecture and places an emphasis on the process of architecting in general, and more specifically on the development and improvement of the architecture function. DYA was developed from the author's experience that the bottleneck for the core processes of enterprise architecting is not the design of architecture products, but rather the embedding of those products in the organisation. DYA is founded on a number of principles, which can be summarised as:

- The enterprise architecture process is as important as the enterprise architecture products. The objective of an architecture should not be to deliver architecture products, but rather to support the enterprises change processes.
- Just enough, just in time architecture. An enterprise architecture can be implemented step by step, driven by business needs. It is not necessary to design all enterprise architecture products in one step.
- Deviations from the architecture are allowed on occasion, but only in a controlled way.

An essential element of DYA is the DYA model (represented in Figure 5.3). This model contains four processes which cover the enterprise architecture core processes. The Strategic Dialogue (between business and IT management) is concerned with the translation of the business strategy into project proposals. By default projects are realised using the enterprise architecture (Development with architecture), but situations may occur in which projects are realised without conformance to the enterprise architecture (Development without architecture). Within this model, the architecture services processes for defining and managing the architecture are positioned as supporting processes. DYA does not prescribe any methods or techniques on how to develop enterprise architecture products.

As of version 3.0, Capgemini's Integrated Architecture Framework (IAF) [30] makes a distinction between the content of architecture and an architecture process. IAF states that an architecture process should always be tailored to a specific situation, characterised mainly by its leading purpose. For the architecture process, IAF gives, analogous to the alignment perspectives as first introduced in [99] and further elaborated in [55], several "architecture process patterns" called *roadmaps*. IAF gives three examples of such roadmaps, namely (1) integrated top-down route, (2) IT-focused route, and (3) problem-focused route. The *integrated top-down route* roadmap (see Figure 5.4) is mainly used to integrally redefine the structure of business, information, information systems and technology infrastructure, as a means to steer an business/IT transformation. This roadmap is characterised by a top-down pattern, mutual iterative refinement between a broad range of aspect areas and an integrated base-line at the logical level before proceeding to the physical level. In the *IT-focussed route* roadmap (see Figure 5.5) the focus is on understanding the business and information to translate them into architecture for information systems and technical infrastructure. Compared with the integrated top-down route – because of its limited scope – it will be faster, more focused on IT and less complete on the busi-

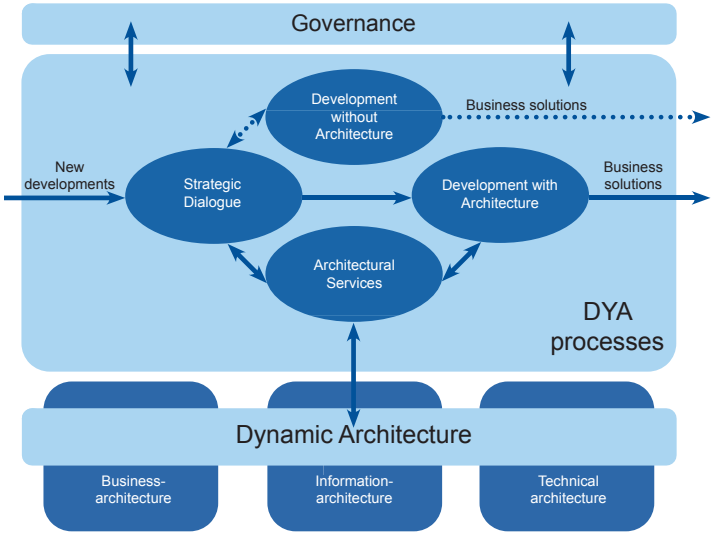

Fig. 5.3 DYA model

ness and information side. Therefore it will typically be useful to support IT-related decisions, accepting the business and information as a given. The *problem-focused route roadmap* (see Figure 5.6) is used to determine the impact and consequences of changes in (the environment of) some current system, so at the physical level. If one understands the consequences at a lower abstraction level and translated them into effects on a higher level, management is able to take measures (bottom-up). These measures at a higher level can then again be translated into principles or guidelines, to be applied at the lower level (top-down).

Applying the IAF-terminology to our focus, the process for enterprise architecture, we could discern a *typical EA-roadmap*, which gives a first elaboration of the overall strategy of the enterprise, clarifying e.g., what and how centrally managed, what are major guiding principles, what needs to be shared and what freedom is left to the lowest levels of the organisation. Such a typical EA roadmap will generally address all aspects areas (Business, Information, Information System and Technical Infrastructure), concepts and services on contextual and conceptual level; principles, guidelines & standards and recommendations for transformation. Where the first two roadmaps can be directly applied to common EA-purposes in supporting decision making, the problem-focused route roadmap can be used to create awareness for enterprise architecture.

IAF focuses on content and the process to create the results. Relatively much freedom and flexibility is given on how to shape and execute the architecture process.

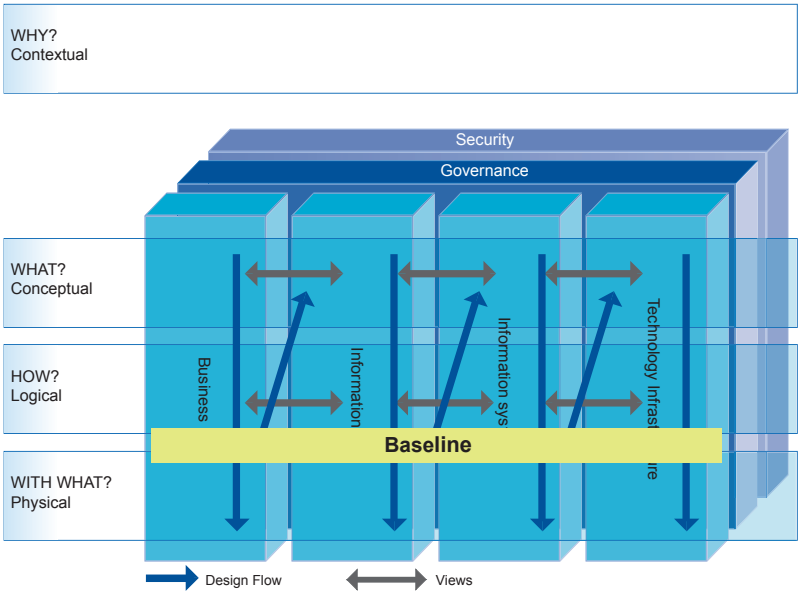

Fig. 5.4 Integrated top-down route roadmap

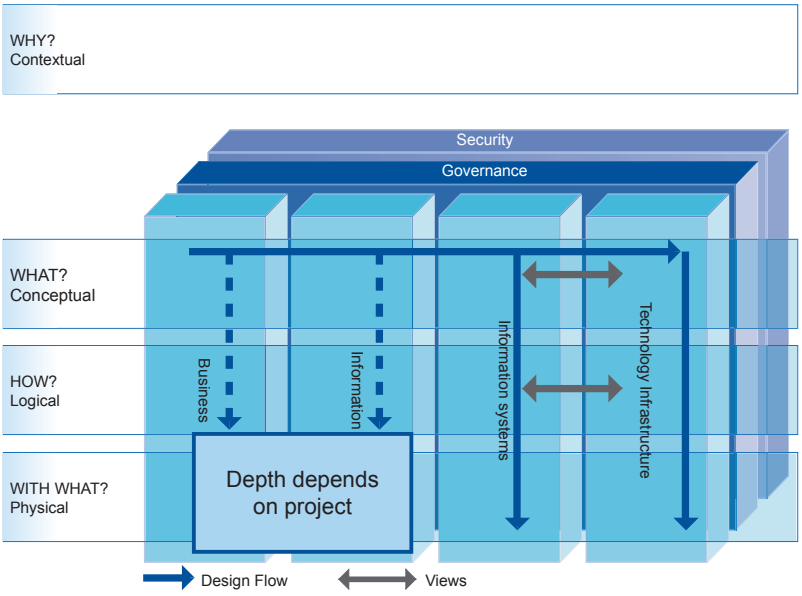

Fig. 5.5 IT-focussed route roadmap

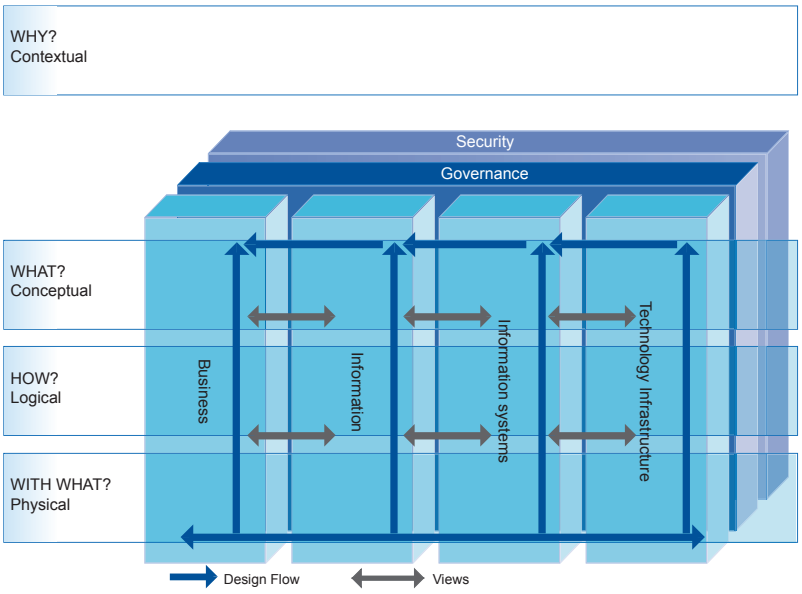

Fig. 5.6 Problem focussed route

5.3.2 Architecture schools: check the "Instruction for use"

As mentioned before, several approaches with a proven track record for the core enterprise architecture processes have emerged, some even taking the form of an architecture school or style. Within each of these schools, we see that the process of creating an enterprise architecture has been elaborated best. The processes involved in for the application and maintenance of an architecture are at best mentioned, but are in need of further elaboration.

We believe that there is a striking resemblance between enterprise architecture schools and approaches to strategy formation. In [85] ten different approaches to strategy formation are described. The authors of [85] argue that each of these approaches or schools have specific contributions and limitations. Some of the schools are prescriptive in nature, concerned with the formulation of a strategy, thus emphasising its content. Other schools describe how a strategy is produced, paying less attention to prescribing the ideal strategic behaviour, thus emphasising its process. As examples of a process school, the *learning school* beliefs that strategy formation is an emergent process in which strategies can only be developed in small steps as an organisation learns over time. Another process school, the *power school* treats strategy formulation as a process of negotiation between conflicting groups within an organisation. A typical example of a content school, the *design school*, states as essence of a good strategy a created fit between internal capabilities and external possibilities, preferably expressed as a Strength-Weakness-Opportunities-Threats (SWOT) analysis.

We see similar characteristics in enterprise architecture schools. Some architecture schools or enterprise architects place emphasis on the aspect of communication with stakeholders. In their view the creation and application of enterprise architecture is only successful if all stakeholders are involved in the process and have expressed their support for the results. Emphasis is put on reaching a shared conceptualisation and less emphasis is put on the quality of the tangible enterprise architecture results themselves. Risks that may arise from these architecture schools are for example: a communication and change process just for the sake of it, the lack of sufficient detail in the results to guide the transformation or even enterprise architecture results that are developed only for shared conceptualisation and not for decision making. Other architecture schools or enterprise architects put an emphasis on tangible enterprise architecture results, often prescribing what method and techniques to apply in developing certain enterprise architecture models. Some potential risks associated with this type of architecture school are: not enough attention for stakeholders, models for the sake of models, too detailed models or focus only on the creation of the enterprise architecture.

Each of these architecture schools have a specific perspective that focuses on one or more specific aspects of enterprise architecture. Each of these perspectives is valuable in itself, providing interesting insight into aspects of enterprise architecture. Each of the architecture school therefore has its value, given that we recognise their difference and underlying premises. Caution, however, must be taken to avoid the pitfall to believe that a single school is apt and applicable to all situations. Note that some have the tendency to promote their architecture school almost as if it is a "religion" to be applied to all situations and that discussing an alternative approach to enterprise architecture might easily lead to an argument.

Before adopting an architecture school, one should assess whether the underlying premises and principles of that school are applicable to the situation at hand. Ideally, every architecture school should contain an instruction for use, explicitly indicating those premises and principles, the intended value, the situations in which (not) to use, risks and the requirements on the enterprise architect and the organisation. Unfortunately, these instructions for use are hardly documented.

5.4 Architecture effectiveness and organisational context

At the beginning of this Chapter, we described the core enterprise architecting processes *create*, *apply* and *maintain*. Not every organisation, which utilises architecture as a steering instrument, should put equal emphasis on each individual architecting process. By recognising architecture maturity levels, organisations are able to position their own architecture efforts and are able to define their architecture process needed. In this Section we discuss an example architecture maturity framework and an example approach for assessing the effectiveness of an organisation's architecture function.

5.4.1 Architecture maturity level of the enterprise

To measure architecture maturity, architecture (capability) maturity models (AMMs or ACMMs) are created. AMMs are based upon capability maturity models [59] that are formal ways to gain control over and improve architecture processes as well as to assess organisation's development competence. Several AMMs exist, for instance the USA Department of Commerce (DoC) ACMM [33] which provides a framework that represents the key components of a productive (IT) architecture process, models that are linked to the Balanced Score Card [71] concept [51]and models for extended-enterprise-architects [120]. All these models have five or six levels of maturity that vary from *initial* to *optimised*.

In this book we use the DoC ACMM as a basis for an AMM since it is widely accepted in the market and it is, in our opinion, not limited to IT architecture. The DoC ACMM contains six maturity levels (see Figure 5.7). Each level has its specific characteristics:

Level 0: None – *No architecture program and no architecture to speak of.*
At this level an organisation does not have an explicit architecture. Most of the times, only an implicit architecture with a small scope exists, covering only few (most of the times only IT related) aspect areas. In the dialogue with stakeholders, one can use a bottom-up approach by communicating about the successes of creating and using the implicit architecture or a top-down approach by communicating the intended value of architecture for the organisation.

Level 1: Initial – *Informal architecture process underway.*
This level is characterised by ad hoc and localised processes, no unified architecture process across technologies or business processes, little or no adherence to existing standards, limited management team awareness or involvement in the architecture process and no explicit governance of architectural standards.
At this level, there will be only one (or few) stakeholder(s) who will see the added value of architecture and architecture capabilities only exists with a limited number of people. Enterprise architecture is not embedded in the corporate governance processes and the emphasis in the communication to stakeholders should be in terms of the benefits of architecture and making them enthusiastic. The focus will be on the *creating* process of enterprise architecture.

Level 2: Under Development – *Architecture process is under development.*
At this level, architecture process has developed clear roles and responsibilities, vision, principles, business linkages, baseline, and target architectures are identified and architecture standards exist, but not necessarily linked to to-be architectures. Reference models and standards have been established. There is an explicit linkage to business strategies, management awareness of architecture effort and governance of a few architectural standards and some adherence to existing standards.
At this level architectures exists and the focus lies on *applying* them. Architecture products are being used in individual decisions and embedded in some

decision documents. The focus in the communication should be on making all relevant stakeholders aware on the added value of enterprise architecture.

Level 3: Defined – *Defined architecture including detailed written procedures*.
At this level the architecture is well defined and communicated to staff and business management with operating unit responsibilities. The process is largely followed, gap analysis and migration plan are completed and architecture is integrated with capital planning and investment control. Also the senior management team is aware of and supportive of the enterprise-wide architecture process and most elements of operating unit show acceptance of or are actively participating in the architecture process.
At this level architects will be consulted as content experts for advice, but don't have authority to enforce compliance to the enterprise architecture.

Level 4: Managed – *Managed and measured architecture process*.
At this level the architecture process is part of the culture. Quality metrics associated with the architecture process are captured, architecture documentation is updated on a regular cycle to reflect the updated architecture, senior management team directly involved in the architecture review process, the entire operating unit accepts and actively participates in the architecture process and formal processes for managing variances feed back into the architecture.
At this level the *maintain* EA process has been implemented and the authority of architects has increased: architecture is embedded in overall governance and decision making processes. EA is used to make informed decisions at senior management level in stead of architecture level.

Level 5: Optimising – *Continuous improvement of the architecture process*.
This level is characterised by concerted efforts to optimise and continuously improve architecture process, architecture process metrics are used to optimise and drive business linkages, business is involved in the continuous process improvements of the architecture, senior management involvement in optimising process improvements in architecture development and governance and feedback on architecture process from all operating unit elements is used to drive architecture process improvements.

Depending on the maturity level, the enterprise will be familiar with the usage and benefits of architecture. The higher the level, the higher the acceptance of architecture as a means and less focus on marketing of enterprise architecture or the process of designing an enterprise architecture is necessary. Each level has its specific subjects to manage: on the lower levels the emphasis is on managing architecture awareness within the organisation, architecture skills and architecture processes. On the higher levels the emphasis is on managing the architecture results, participating in transformation steering, informed decision making and continuing improvement of the architecture function within the organisation. At the lower levels the focus will be on the *creating* enterprise architecture process, while at the middle levels the *apply* enterprise architecture process will be introduced followed by the *maintain* enterprise architecture process at the higher levels.

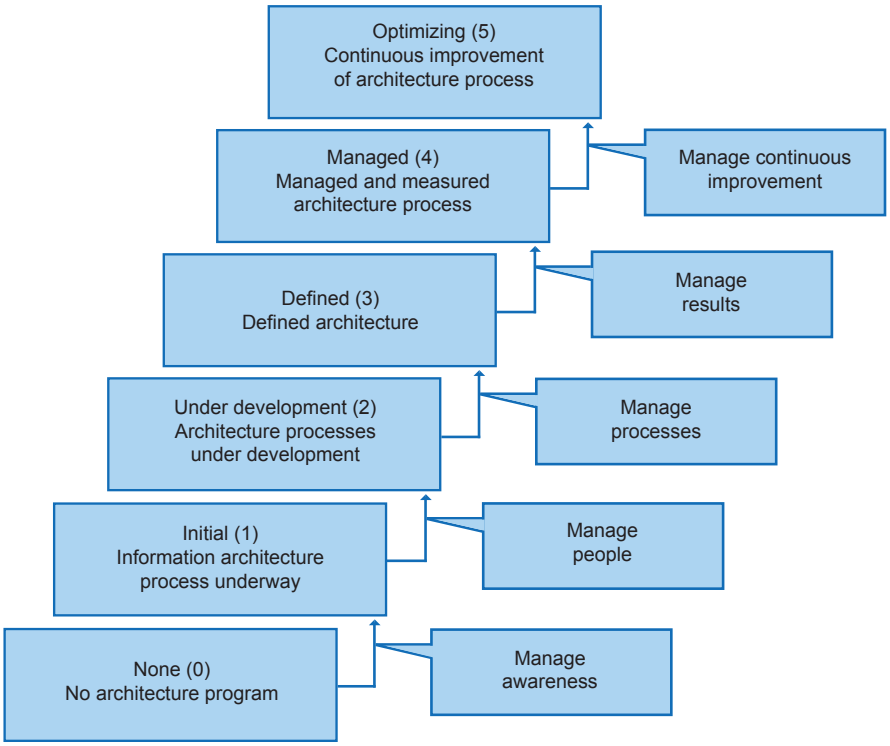

Fig. 5.7 Architecture maturity model

5.4.2 Assessing an organisation's architecture effectiveness

In this Subsection we focus on the effectiveness with which an organisation is engaged in enterprise architecting. The discussion below is strongly based on NAOMI (Normalised Architecture Organisation Maturity Index) as reported in [112]. The NAOMI assessment approach is designed to determine an organisation's architecture effectiveness. It judges an organisation on its ability to reach the goals it set with architecture. NAOMI provides three main assessment variables, which provide three different perspectives on architecture effectiveness:

1. architecture awareness;
2. architecture maturity;
3. architecture alignment.

Enterprise architecture typically starts with an immature level of architecture awareness. The origin of architecture awareness might differ per organisation. It might be the board members initialising an architecture program in order to cope with business complexity, or to guide a large business transformation. On the other

hand, it might also be a few members of the IT department that introduce IT architecture in order to guide software development projects. From this initial, immature, level an organisation should improve its level of architecture awareness. Indications of a high level of architecture awareness are [111]:

- A clear vision of what architecture is, what the objectives are with architecture, and how these objectives are to be realised; the vision, mission and strategy the architecture function has with architecture;
- A clear view on issues such as how architecture should be applied, which aspects of enterprise architecture they see as vital for the success of the architecture program, and how a positive attitude towards architecture is to be created within the affected organisation;
- A clear description of the architecture processes and work instructions.

Architecture maturity indicates how well the architecture function puts architecture into practice. Compared to architecture awareness, this perspective focuses on how well architects succeed in the realisation of architecture instead of how architecture is experienced in the minds of architects. Architecture maturity is about the actual observable behaviour of the architecture function.

An organisation may have several architecture functions, with a focus on different types of architecture. It is key to align those by ensuring the responsible architecture functions cooperate. Such collaboration indicates the level of architecture alignment.

Figure 5.8 shows the structure of NAOMI. A single-headed arrow indicates that the source variable explains the destination variable. A double-headed arrow indicates that the two variables correlate. All six intrinsic variables (Governance, Processes, Communication, Support, Scope, Resources) explain each other, since they all depend on each other. Please note, however, that Figure 5.8 shows only lines from the six intrinsic variables to the three key variables in order to keep the Figure simple.

In order to determine an organisation's levels of architecture awareness, maturity and alignment, NAOMI uses six underlying intrinsic variables:

Governance – This represents the managerial and organisational aspects of enterprise architecture. An architecture function, as well as any other business unit or department, needs to create its own vision, mission and strategy. By doing this, the architecture function states its role and justification of its existence within the organisation, its added value and strengths, its strategic objectives, and the direction in which it wants to reach those objectives. This allows the architecture function to have a clear focus, which should be aligned with the overall corporate (business and/or IT) strategy. Based on its strategic objectives, an architecture needs to create an internal organisational structure, and needs to plan its activities in order to reach those objectives.

Processes – An architecture function should clearly describe its primary and secondary processes. Primary processes involve the development, maintenance, and implementation of architectures. Secondary processes entail architecture

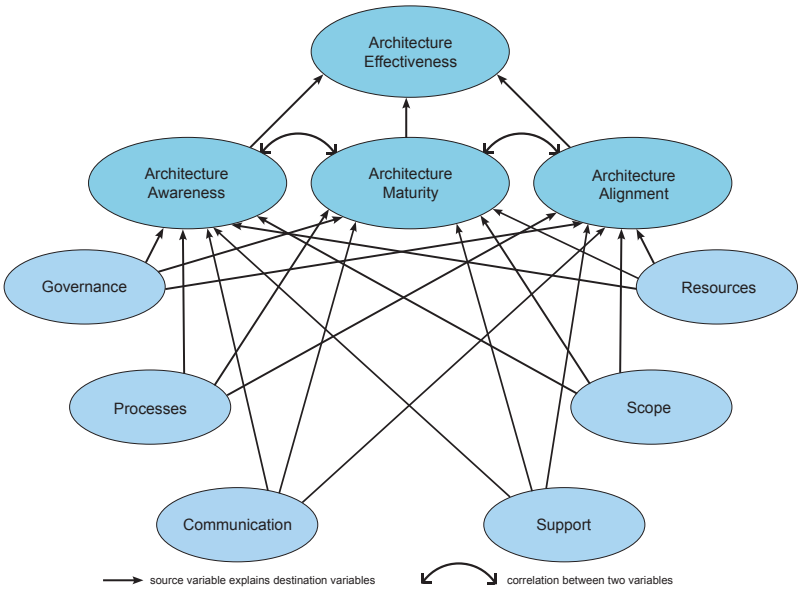

Fig. 5.8 Structure of NAOMI

knowledge and quality management, which focus on improving the quality and efficiency of the primary architecture processes.

Communication – The level in which an architecture function is able to communicate with its stakeholders through architecture is essential in determining its ability to be effective. However, another important issue is how an architecture function is able to communicate about the architectures it creates.

Support – An architecture function can only successfully put the architectures it creates into practice with the support of the rest of the organisation. Essential in determining the level of organisational support for the enterprise architecture program is the level of organisational acceptance of the architecture-driven changes.

Scope – The organisational scope of architecture indicates which part of an organisation (which departments, units, divisions) is involved in the enterprise architecture program. The percentage of all departments, business units, or divisions that work according to architecture indicates the broadness of the architecture program. The broader the architecture program, the higher the organisational impact will be. The type of departments – business or IT – where architecture is being used determines the organisational emphasis of the architecture program.

Resources – The resources an architecture function needs for developing, maintaining and realising its architectures are twofold. Firstly, it needs human resources – e.g., IT architects, business architects, an architecture manager. Sec-

ondly, it also needs frameworks, methods, techniques and tools, which provide a standardised way of working.

In general one can state that at the lower maturity levels, organisations score low on all NOAMI variables. At the middle maturity levels the score on the variables *processes*, *communication* and *scope* increases while at the highest maturity levels organisations enhance their score on all variables of the NAOMI model.

5.4.3 Assessing an organisation's culture and management style

Organisational culture and *management style* will also be influencing factors on the process. These factors will impact the way how to best cope with stakeholders and how to get them involved in and committed to the process and enterprise architecture results. In an organisation which has a culture of consensus decision-making and therefore many stakeholders which need to be involved, typically the enterprise architecture process will only be successful if a number of workshops with all stakeholders are held to get them involved, to reach consensus and to make decisions. As another example, take an organisation where the most powerful person (e.g. the founder and owner of an organisation) or group (e.g. marketing) decides and others will follow. If this is the case, the process will need to include dedicated activities for communication of the decisions with the others. Organisational culture and management style not only have an influence on the creation of an enterprise architecture, but also on the application and maintenance of an enterprise architecture. As a last example, in an organisation where freedom is valued more than compliance to company rules, using an enterprise architecture to restrict design freedom of individual projects can only be successful if proper attention is given to communication and compliance monitoring.

5.5 Organising the architecture function

We now turn to the question how to get architecture work moving, to keep it working and how to continuously improve it. In Section 5.2 we introduced the creating, applying and maintaining of enterprise architecture as the "core process of enterprise architecting". To get the *act* of this "core process" running, we need to *plan* it and to *learn* from it. We define the *architecture function* as the *PLAN-ACT-LEARN cycle* for the creating, applying and maintaining of enterprise architecture. The PLAN contains the planning of all architecture work (creating, applying and maintaining); it serves to optimally use time and resources of the enterprise architecture team and the stakeholders. In the ACT the core of the architecture work we described in 5.2 is executed according the PLAN. In the LEARN we take lessons from all aspects of the architecture work done, varying from the effectiveness of the enterprise architecture and the quality of architecture results via efficiency of the architecture process to the

causes of all that in organisation, involvement of stakeholders and the competencies and capacity of the architecture team. By a well-implemented LEARN the wheel of continuous improvement to enable the quality of architecture work will get running. The thus defined architecture function needs to be ORGANISEd, i.e. embedded in the organisation with tasks, responsibilities and authority and implemented by people and means. From the PLAN-ACT-LEARN-ORGANISE (PALO), we will now describe the PLAN, the LEARN and the ORGANISE; the ACT has been treated already in Section 5.2.

5.5.1 PLANning activities

By the PLANning activities for enterprise architecture, we understand the activities to continually understand the context of controlled change, the potential contribution of enterprise architecture to the control of this change – so the requirements on the enterprise architecture –, and the approach answering this need, setting the agenda for further activities.

The PLAN contains all preparatory work for the architecture function, in cooperation with its stakeholders, to ensure the architecture work becomes manageable. It typically results in Plans of Approach (PoA) for creating, applying and maintaining, including the way of working (e.g. workshops, interview), the time to be spent and whom to involve. Table 5.1 shows some examples of the planned activities and what the PoA states about these activities.

5.5.2 LEARNing activities

In the LEARNing activities for enterprise architecture (1) lessons are distilled from change processes to further improve the content of the enterprise architecture, the enterprise architecture capabilities and their effects on controlling change, and, based on that, (2) proposals for improvement are concluded, e.g. to more effectively involve stakeholders or to make the next maturity-leap. This learning process is by no means linear; all stakeholders learn, develop and choose while participating in the architecture process. The learning comprises again creating, applying and maintaining. For each of those we will now give examples of learning experiences and improvement proposals:

- LEARN on creating:
  - let's record the updated architecture results in a formal architecture tool;
  - no common framework of reference between the architects is in use; let's adopt a common language for architecture results & process;
- LEARN on applying:

P-activity	Example planned activity	Parts of PoA for example P-activity
P-create	To deliver before the end of the next fiscal year a separate architecture study for the newly acquired organisation, including synergy opportunities with our existing organisation	From the PoA: 5 milestones each culminating in a one-day workshop; enterprise architecture study to be executed with staff from both organisations (own, plus newly acquired); first workshop is the architecture method adoption workshop, to be attended by all architects involved
P-apply	Deliver an application portfolio rationalization proposal to save 30% annual maintenance costs, based on the current architecture models, the formulated business- and IT-strategies	Use questionnaire to build the application inventory and record it in the architecture tool; build business services model & validate this with the business; map application portfolio on business services, detecting gaps and overlaps; let business owners present rationalization proposal to CEO/CFO/CIO
P-maintain	Monitoring relevant external and internal events, resulting in a (1) quarterly newsletter for stakeholders and (2) a yearly one-day-workshop to translate this into a new yearly plan	Each architect may invest 10% of his / her time; Architecture Program Mgt Office produces newsletter; invited stakeholders for yearly workshop involving senior management of all business units, the CIO and CFO

Table 5.1 Example fragments of PLANned architecture activities

  - improve boardroom communication skills;
  - let's design and agree on an exception handling process, when the design team and architecture team cannot reach agreement;
  - our procedure for assigning building permits does not work; reframe the procedure to a more collaborative approach – architecting *by walking around*;
- LEARN on maintaining:
  - install a classification for defect reporting;
  - monitoring was so much focused on the business, that technology developments escaped from our attention;
  - project results haven't consequently been fed-back to the enterprise architecture team; let's assign project budget for that and make it a condition for sign-off.

5.5.3 ORGANISing architecture activities

To get and keep the described cycle of Plan-Act-Learn activities running, the architecture function needs to be ORGANISEd, i.e. embedded in the organisation with tasks, responsibilities and authority and implemented by people and means. Of course this is an example of more generic organisation design, as treated in [7] and [124]. We will therefore restrict ourselves to some typical example results in

organising enterprise architecture activities, namely in assigning responsibilities, the use of certain (architecture process) principles, the use of KPI's and CSF's, the use of out-sourcing/off-shoring, the building of competencies and the use of tools. Generally speaking, we expect that in a more mature organisation also more explicit attention for implementing the architecture function will be given. At a basic level of maturity the way of working will be typically project-bound, while at a higher level of maturity the way of working will be more structural, continuous and embedded in the over-all governance of the enterprise.

A typical result of organising would be a table for Responsible, Accountable, Consulted and Informed roles, the so-called RACI-table. Example expressions from such a RACI-study could be:

- e.g., lead architect reports to CIO council,
- e.g., separate architecture team / let architecture be part of the over-all governance,
- e.g., for establishing technology standards a CTO needs to be consulted,
- e.g., in our architecture team we discern formal roles of solution architect and enterprise architect, from which we expect the following competencies (using the *Competency language* from Chapter 6).

Another typical result of organising enterprise activities would be to produce a list of supported *architecture process principles*, containing e.g.:

- comply or explain,
- independent judgments can only be made by external parties,
- all investments exceeding the EU-tender threshold (2008: €133.000 for deliveries and services) have to be co-signed by the Corporate Architect,
- enterprise architecture we do ourselves, solution architecture is out-sourceable,
- (as suggested by L-maintain) each project should include budget for feeding-back project results into the enterprise architecture; no project sign-off will be given without this feed-back,
- each project is assigned an architect, paid by the corporate architecture budget,
- the sise of the architecture team should be at least 0,2% of the total workforce of the enterprise.

An organised professional architecture function will also state its Key Performance Indicators (KPI's) and its Critical Success Factors (CSF's); examples of that could be:

- KPI-example: customer satisfaction for architecture work should be 7,5 on a scale of 1 to 10 (1=bad, 10=excellent),
- CSF-example: 6% – 10% from the corporate architecture budget should be spent in programmes and projects.

It is important to organise according to the current and aspired maturity-level. A possibility for that is to make this organising a yearly process, profiting from the LEARN-results. Suppose the enterprise is executing a change process from roughly AMM-level 2 to 3, enabling the corporate use of enterprise architecture, impacting

the over-all governance. Then it would be to reserve for this year part of the corporate architecture budget for describing internal, governing and complying processes.

A special question for ORGANISing is the outsourcings question: what part of the architecture work is out-sourceable / off-shorable / shareable to what party. To answer this question, it would help to discern e.g.:

- business architecture, information architecture and ICT architecture work;
- enterprise architecture and solution architecture;
- steering the architecture process and executing the architecture process;
- type of competencies required and available (see Chapter 6);
- type of architecture results needed (see Chapter 4);
- the phase of maturity (AMM), e.g. in AMM-phase 0/1 input by external architects generally will speed up the learning process;
- the degree in which the business processes itself will be outsourced.

In the latter case, enterprise architecture itself will be used to support and monitor compliance between own enterprise processes and outsourced processes/services. Enterprise architecture is prescriptive for the outsourced services, i.e. it is up to the supplier of outsourced services to design and implement their solution within the guidelines of the enterprise architecture, e.g. with respect to reliability or flexibility.

Returning to the outsourcings question for the architecture function in general, we note the following. On one side we see work which should, according to best practice insights, probably stay with the company itself, such as enterprise architecture, business architecture and steering the architecture process. On the other side we see work which is a more likely candidate for outsourcing, e.g., IT architecture, solution architecture and executing the architecture process. And of course this outsourcing question for the architecture function is a specific case for outsourcing strategies in general, as studied by [48, 81].

To a large extent, the success of the enterprise architecture process will be determined by the team that executes the process. In the next Chapter, we will therefore not only discuss the competencies and responsibilities of enterprise architects, but also elaborate on their role in teams [16, 26]. Here we will briefly highlight some factors that apply specifically to *successful enterprise architecture teams* and their impact on the architecture processes:

- An organised team with clearly defined roles and responsibilities, sharing a clear purpose and goal for the architecture work.
- An agreed common language and way of working. This might lead to the selection, adoption and tailoring of an architecture framework, methods, techniques and tools and/or training the team members.
- The right skill sets and competencies (to be discussed in Chapter 6), possibly involving hiring, training and coaching of team members.

The enterprise architecture processes will be more efficient when supported by *tools*. The same, however, holds true for tools as for the process: there is no one-size-fits-all set of tools that is appropriate for all situations. Tools can range from simple, general purpose office tools to specialised enterprise architecture tools, useful for

developing, applying and maintaining architecture products. At higher levels of architecture maturity, where the continuous use of enterprise architecture is supported, it is worthwhile to implement and use specialised enterprise architecture tools. At lower levels of maturity, the focus should be on people. Do not expect the enterprise architecture tools to do the job.

Other tools to be considered are tools to facilitate communication and decision-making. Group decision support systems for instance can be used to support complex decision making by a large group of people [73, 89, 144, 143, 146]. Using these types of systems, the opinions of (groups of) stakeholders can be systematically collected, ordered, evaluated, discussed and reported. As another example, *business incubators* can speed up the drafting of plans, decision-making and the creating of commitment for all parties involved. It is important to relate the use of these and other tools to the preferred and feasible way of decision-making.

To grow in quality as architecture function, a number of quality frameworks can be applied, such as from Six Sigma [110] and EFQM [54]. We now take as example the EFQM Excellence Model, as described on [38] (see Figure 5.9, taken from [38]):

> *The EFQM Excellence Model is a non-prescriptive framework based on nine criteria. Five of these are 'enablers' and four are 'results'. The 'enabler' criteria cover what an organisation does. The 'results' criteria cover what an organisation achieves. 'Results' are caused by 'enablers' and feedback from 'results' help to improve 'enablers'. The model, which recognises there are many approaches to achieving sustainable excellence in all aspects of performance, is based on the premise that excellent results with respect to performance, customers, people and society are achieved through leadership driving policy and strategy, that is delivered through people partnerships and resources, and processes.*

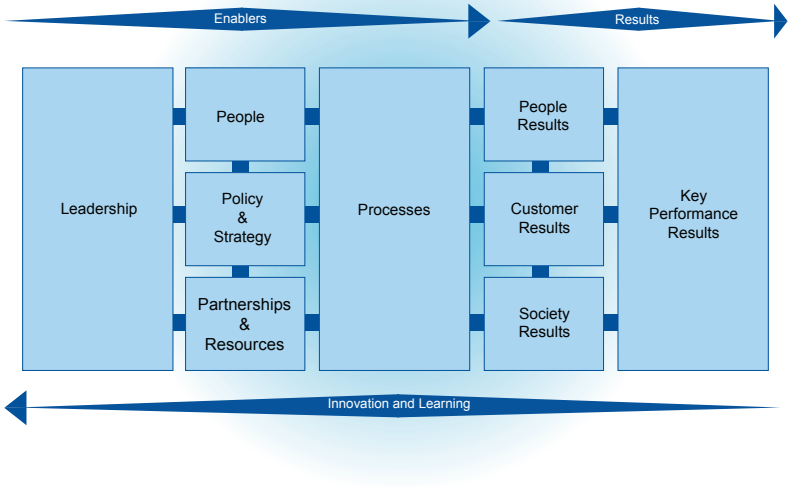

Fig. 5.9 The EFQM Excellence Model

When we compare this with our enterprise work, we see it deals with people (Chapter 6), processes (Chapter 5), results (Chapter 4) and in the end of course effectiveness (Chapter 2 and Chapter 3). In this Chapter, we have also identified the need to learn.

5.6 Summary

As discussed in this Chapter, a good enterprise architecture process depends on a number of situational factors; therefore a one-size-fits-all approach does not exist. It is important to keep in mind that any enterprise architecture is a means to an end; it should deliver value by answering questions of stakeholders. It is essential therefore that the enterprise architecture processes does not solely focus on the delivering the enterprise architecture products, but start from the relevant stakeholders, and understanding their concerns, objectives and stated or implied requirements. In Chapter 4 we already saw that such thinking about purpose, scope, steering needs and expected benefits should determine the architecture results to be delivered. We have seen that this also can influence the architecture process, e.g.,

- to reach agreement or good understanding of scope, purpose, etc., an iterative process might be needed;
- the process around the results will determine the perception of it; even the best results will not work, unless its benefits have been recognised.

Next to this criterion of effectiveness, the architecture process should be efficient, i.e., it should only develop those results (end results and intermediate results) that are necessary to address the concerns of the stakeholders. At the same time the focus in the process should not be on the architecture results, but on the outcomes. An efficient process should therefore include all communication necessary to ensure that results are really applied as intended, including investing time in shaping of relevant views.

Currently, hardly any scientific research or publications exist that describe success factors for enterprise architecture processes. Most of the approaches and architecture maturity models are bundled as best practices of experts. Still, an architecture maturity model can help to cope with architecture maturity and increasing the maturity. The higher level of architecture maturity, the less focus will be on designing and implementing the architecture processes, but more on optimising those processes. This will decide which part of the architecture should get what emphasis; e.g. a valuable policy in phase 0/1 could be “start by a simple result and show to a few IT people”. We note that most of those approaches lack an “instruction for use”, making the (many times implicit) assumptions on usability and feasibility explicit. Also we note that an architecture process really is a business process as many other processes, which evokes the question how AMM, NAOMI and the drafted PALO-notion could benefit from general quality frameworks such as those from EFQM and Six Sigma.

To common descriptions of the architecture process, which tend to emphasise the creating (order of working and products) and sometimes the applying of architecture results, we added maintaining results. Also we argued that this *Create* / *Apply* / *Maintain* activities need not only acting, but also planning and learning. This 3 by 3 matrix *Plan*/*Act*/*Learn* versus *Create*/*Apply*/*Maintain* needs to be *Organise*d, implemented in the organisation. Also for this Organise, experience is still on the level of typical examples, no patterns do exist yet to combine maturity with each PALO-aspect. For example, it would help to connect maturity levels of the organisation to the role of enterprise architecture and enterprise architects in formal decision making, preventing the use of *building permits* or contracts on AMM-level 0. *Learn* has to look next to *Plan*/*Act* also to *Organise*: are we still effectively and efficiently organised and are we delivering the results that our key stakeholder value?

We now summarise the core aspects of the process of enterprise architecting.

- Create:
  - Understand purpose and context;
  - Determine deliverables;
  - Monitor context and stakeholders;
  - Create shared conceptualisation;
  - Design creation process;
  - Determine impacts;
  - Communicate.
- Apply:
  - Inform;
  - Support decision-making;
  - Ensure compliance;
  - Make results available;
  - (Re)-communicate.
- Maintain:
  - Monitor context and stakeholders;
  - Assess drivers for change;
  - Update and (re-)communicate.
- Organise:
  - Organise team;
  - Select frameworks, tools and tricks;
  - Communicate about enterprise architecture as a means;
  - Embed enterprise architecting in governance;
  - Monitor maturity;
  - Manage quality;
  - Establish leadership;
  - Innovate.

5.7 Discussion statements

1. Enterprise architecting will only succeed if the architecture is just in time and just enough.
2. With change being a constant, a stable enterprise architecture that will last more than three months, is an illusion. Maintenance of the enterprise architecture will be the core process.
3. There is no one single way of doing enterprise architecture.
4. Each school of architecting has its own virtue.

Chapter 6
The Enterprise Architect

6.1 Introduction

In the previous Chapters we have discussed the concept of enterprise architecture, the deliverables produced during enterprise architecting, as well as the processes involved. We have not yet discussed the person who needs to execute these tasks; *The Enterprise Architect*. In this Chapter, which is largely based on [130], we aim to discuss the competencies, attitudes, and abilities needed by an enterprise architect to best conduct their tasks.

One only needs to look at one of the many job-ads to see that an enterprise architect needs to have a wide range of competencies. Consider for example:

> *Assist the Enterprise Architecture team to develop a Target EA, Transition Plan and EA governance strategies. Work with lead to develop all stages of enterprise architecture, information engineering, system development methodologies, EA strategic planning, business process re-engineering, workflow processing, requirements analysis, prototyping, system testing, major system and database implementation. Assist in the development of an EA roadmap and strategy, current architecture assessment, architecture tools and repository evaluation and approach, development of EA governance, communication, metrics, investment management, modelling of current and target architecture views, gap analysis, and migration plan to integrate their IT efforts with mission goals.*
>
> From: `http://hotjobs.yahoo.com/jobseeker/jobsearch/job_detail.html?job_id=JVVWL53A4E1`

This example shows that the role of enterprise architect demands leadership qualities, a deep knowledge of IT and business domains as well as ample communication skills. Clearly not a starter's position.

In randomly chosen job ads for enterprise architects, the following tasks and responsibilities are asked for:

- *Responsible for executing the architectural vision for IT systems within the organisation including those that support Internet applications, ensuring that architecture conforms to enterprise standards.*
- *Provide technical and architectural direction to the software and infrastructure team.*

- *Stay constantly attuned to emerging technologies and recommend business direction based on those technologies.*
- *Provides technical expertise to peers and associates on overall distributed enterprise architecture and design.*
- *Assist in developing and maintaining strategies that result in efficient and effective use of enterprise core services.*
- *Strong conceptual and analytical skills.*
- *Experience in creating and defining new technology concepts and solutions.*
- *Java development experience preferably in SAP Enterprise Portal environment.*
- *Experience in development of Segment Architectures that align with and enable agency strategic goals and business requirements.*

The requirements put on an enterprise architect seem to range from very specific programming skills to broad leadership qualities as well as the ability to develop a business strategy. Tasks and responsibilities differ per job add: there is no one set of tasks and responsibilities for the role of enterprise architect.

Besides an *enterprise architect*, there are many other types of architects, such as business architects, information architects, IT architects, domain architects or solution architects. The difference between these types of architects and the enterprise architect is that the enterprise architect covers the breadth of business and IT, while domain architects focus on one aspect of the enterprise (business, IT, information) and solution architects on one small part of the implementation of the architecture (applications, software, business processes).

Some initial work has already been done regarding the abilities and competencies that should be met by enterprise architects. For instance, organisations such as TOGAF [139] and the Netherlands Architecture Forum [131] have created frameworks of competencies for architects. Some organisations have created their own competencies frameworks [2, 150], or have even introduced their own certification programs (for example: IBM, HP, Capgemini, Federal Enterprise Architecture Certification Institute, and TOGAF).

Standard guidelines regarding the competencies of an enterprise architect still lack. Responsibilities differ per company/assignment and research showed that architects themselves expect to have to have a variety of competencies [145]. Using pre-existing frameworks for competencies and abilities [16, 26, 2, 29, 131, 139, 145, 150] as a starting point, this Chapter provides a competency framework for enterprise architects which is geared towards the responsibilities of enterprise architects. The latter responsibilities are derived from the enterprise architecting processes (and associated tasks) discussed in the previous Chapter.

This Chapter is structured as follows. In Section 6.2 we discuss the basic competencies which an enterprise architect is expected to have. Section 6.3 summarises the responsibilities of enterprise architects based on the processes discussed in the previous Chapter. Section 6.4 then continues by discussing the personality types needed to meet these responsibilities. Since enterprise architects are likely to operate in teams, Section 6.5 considers competencies related to working in teams. Finally, before concluding this Chapter, we will briefly visit the topic of professional

development of the enterprise architect, which is about cooperation and sharing best practices and certification.

6.2 Relevant competencies

In this Section we look at the competencies that are relevant to the work of enterprise architects. As we will see in the next Section, not all of these competencies are relevant to each of the roles played by architects.

According to a survey among architects, one has to be a jack-of-all-trades to be a good architect [145]. Even more, job ads for enterprise architects typically claim at least five years of experience, profound domain expertise, specific knowledge about networks, applications, operating systems, etc, communication skills and proven success in implementation. Providing a complete list of competencies of the enterprise architect is therefore also hardly possible. We will limit ourselves by introducing the essential competencies on the different fields which are needed. In doing so, we distinguish two kinds of competencies:

Professional competencies – Competencies dealing with knowledge, attitude and skills necessary to a successful performance in a specific function or role [17].

Personal competencies – Competencies that can be used in several functions or roles (i.e. communication skills) and personality characteristics.

6.2.1 Professional competencies

The professional competencies comprise the knowledge, attitude and skills to perform successfully in a specific function [131]. The enterprise architect should be able to understand and have knowledge of all four areas (business, information, information systems and infrastructure), while he needs to be an expert in at least one area [2]. TOGAF divides the professional competencies in their Architecture skills framework in business skills and methods, enterprise architecture skills, programme or project management skills, IT general knowledge skills, technical IT skills, and legal environment [139].

When looking at the competence model of a standardisation effort such as TOGAF as well as the competence model of an architecture society such as NAF, one can conclude that architects need to have knowledge about the different domains they act in. In addition, knowledge about architecture principles, architecture frameworks and governance is most important, while keeping informed about new developments is also necessary.

6.2.2 Personal competencies

For the personal competencies we do not distinguish between different types of architects. Even stronger, it seems those competencies are quite close to adjacent professions such as strategists, process developers and system developers. The personal competencies can be divided in intermediary competencies, values, norms and ethics and personality characteristics [131]. This last group contains natural abilities of a person and these are therefore hard to be learned. One of these is persuasiveness, which is recognised by [26] as an important characteristic of an architect. Others are independence, persistence, initiative, etc [131]. Values, norms and ethics differ per person and organisation. Intermediary competencies are the ones mostly mentioned in literature and job ads. A short comparison between four sources [14, 131, 139, 151] showed the following top five intermediary competencies for the architect, according to the naming conventions of [131]:

- Analytical skills.
- Communication skills.
- Negotiation.
- Abstraction capacity.
- Sensitivity and empathy.

Besides those, creativity and leadership appear to be essential for the enterprise architect, especially because (s)he needs to cover the whole spectrum of business and ICT and often operates in a leadership role in close collaboration with other architects. Based upon [131] and extended with some competencies concerning change management from [29], we identify the following personal competencies:

Abstraction capacity – The ability to learn in new situations and to adapt acquired knowledge and facts, rules, principles to new domains.

Accurateness – Working neatly and precise.

Analytical skills – The ability to identify a concept or problem, to dissect or isolate its components, to organise information for decision making, to establish criteria for evaluation, and to draw appropriate conclusions.

Authenticity – Being true to one's own personality, spirit, or character.

Consulting – Being able to give recommendations on a certain case.

Creativity – To be able to generate creative ideas and solutions, invent new ways of doing business, and be open to new information.

Decisiveness – To be able to take decisions after having enough or complete information and act towards these decisions.

Dedication – Driven to accomplish their goals.

Didactical skills – The ability to transfer complex knowledge to other people.

Diplomacy – Ability to communicate about sensitive issues without arousing hostility.

Flexibility – Ability to deal with changed conditions, assumptions, environment, etc.

In-dependency – To be able to act without being influenced by others.

Initiative – Readiness to act on opportunities.
Integrity – Moral soundness.
Leadership – Inspiring and guiding groups and people.
Listening – Listen actively to understand information or directions and be able to provide relevant feedback.
Loyalty – Faithful to the key stakeholders.
Negotiation – To be able to maintain a position in conversation with others and improve this position.
Openness – Open to alternative directions, solutions and opinions.
Opinion forming – Being able to make a judgement about a certain case.
Organisational awareness – To understand the inner working of the organisation; to estimate the value of the own influence and consequences of decisions or activities.
Persistence – Being determined to do or achieve something
Persuasiveness – To be able to convince others of a certain opinion
Plan and organise – Making objectives and take actions to reach these objectives in an effective way.
Result driven – To be able to realise objectives and results
Self-confident – Confident about (and familiar with) their own (in)abilities.
Self-development – Reflect on your performance and goals, identify learning needs and development options, and develop knowledge and skills
Sensitivity and empathy – Sensing others' feelings and perspective, and taking an active interest in their concerns.
Stability – Has a stable character and mood.
Teamwork – Working with others towards shared goals and creating group synergy in pursuing these goals.
Verbal communication skills – Use appropriate technical or business vocabulary to be able to express thoughts and feelings in a concise way and to respond adequately to others.
Working systematically – Be able to execute the work in a prescribed way.
Written communication skills – Write clear and accurate reports, letters and documents.

To this list we will add two competencies based upon our own experiences, namely:

Facilitation skills – Be able to facilitate workshops.
Visualisation skills – Be able to visualize architecture results.

6.3 Responsibilities of an enterprise architect

According to [151] an enterprise architect's job can involve governance committees, architecture review boards, technology life cycles, portfolio management, architecture strategy and strategic project support. [26] show that enterprise architecture has

broadened its scope from just an IT issue to the enterprise wide IT architecture and business architecture, with as goal to increase enterprise agility and alignment with business strategy.

Based on the previous Chapter, we identified a number of different responsibilities for the enterprise architect. In meeting these responsibilities, the enterprise architect needs certain personal competencies. Table 6.1 provides a mapping from the responsibilities to the competencies discerned in the previous Section based upon our own experience. We have not mapped the professional competences to the responsibilities, this needs further research.

6.4 Personality types

Strano et al. [132] report on a survey conducted among enterprise architects of the federal government of the United States of America, and concluded that an enterprise architect can have the roles of a *change agent*, *communicator*, *leader*, *manager*, and *modeller*. In [132] these roles are defined as:

Change agent – "*As a change agent, the enterprise architect supports enterprise leaders in establishing and promoting the best strategy to accomplish business goals and objectives.*"

Communicator – "*As a communicator, he assists managers, analysts, systems architects and engineers in understanding the details of the strategy sufficiently well to make decisions and execute the plan that leads to realisation of the shared vision.*"

Leader – "*As a leader, the enterprise architect participates in creating a shared vision, motivating members of the enterprise to aspire to achieving the vision, and providing clear direction regarding what is required to execute a strategy to accomplish goals and objectives that result in performance improvements.*"

Manager – "*As a manager, he organises the architecture team and ensures that adequate resources are secured to perform the architecture process.*"

Modeller – "*As a modeller, the enterprise architect provides a representation of the relationships of enterprise components with sufficient detail and in the format needed to enable making necessary decisions to execute the strategic plan.*"

As an alternative to these roles, [26] suggests four competency areas: credible expert, strategist, politician and leadership. In this Chapter we adapt the roles of [132] since they are based on a documented empirical study.

In [29] five stereotypical styles of thinking about change are identified. Each style is typed by its own-colour:

Yellowprint-thinking – Focuses on bringing interests together, stimulating stakeholders to formulate opinions, creating win-win situations and forming coalitions.

Competences / Architecture Process	Abstraction Capacity	Accurateness	Analytical Skills	Authenticity	Consulting	Creativity	Decisiveness	Dedication	Didactical Skills	Diplomacy	Facilitation skills	Flexibility	Independency	Initiative	Integrity	Leadership	Listening	Loyalty	Negotiation	Openness	Opinion Forming	Organisational Awareness	Persistence	Persuasiveness	Plan And Organize	Result Driven	Self Development	Self-confidence	Sensitivity And Empathy	Stability	Teamwork	Verbal Communication	Visualisation skills	Working Systematically	Written Communication
Create																																		
Understand purpose and context	×	×	×	×	×		×			×				×	×	×	×				×	×				×		×			×	×	×		
Determine deliverables		×	×	×	×	×	×	×	×					×	×	×	×				×	×			×	×					×	×	×		×
Monitor context and stakeholders		×		×	×		×	×	×	×			×		×	×	×	×			×	×				×					×	×	×		×
Create shared conceptualisation	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×	×		×		×					×		×	×		×	×			×
Design creation process	×			×	×	×	×						×	×		×					×					×					×	×		×	
Determine impacts	×	×	×		×																	×											×	×	
Communicate	×				×	×			×	×	×			×	×		×		×	×	×	×	×	×								×			
Apply																																			
Inform	×		×		×				×	×				×						×		×					×	×				×			
Support decision-making	×	×	×	×	×		×			×				×	×	×	×			×	×	×				×		×	×			×	×	×	×
Ensure compliance		×		×	×		×			×				×	×	×	×			×	×	×	×					×	×			×	×		
Make results available	×	×	×	×					×					×						×					×			×	×				×	×		
(Re-)communicate	×		×	×					×					×			×					×				×		×	×							
Maintain																																			
Monitor context & stakeholders		×		×										×			×				×	×													
Assess drivers for change		×												×							×	×	×												
Update & Communicate	×	×	×	×										×		×										×									
Organise																																			
Organise team					×	×	×				×	×		×		×		×	×		×	×		×	×		×	×	×						
Select framework, tools & tricks	×	×												×		×					×	×	×			×		×							
Communicate about EA		×		×	×	×	×	×	×	×	×			×		×	×				×	×	×	×											
Embed EA in Governance		×		×			×	×	×	×	×			×	×	×			×			×	×	×											
Monitor maturity					×	×				×	×			×	×	×						×			×											
Manage quality	×	×		×			×	×	×					×	×	×			×			×	×	×												
Establish leadership		×		×	×	×	×	×	×	×	×	×		×	×	×		×				×														
Innovate	×		×	×	×									×																						

Table 6.1 Competencies mapped upon responsibilities

Blueprint-thinking – Focuses on the formulation of unambiguous objectives, development of a plan of action, monitoring and adjusting the change process accordingly.

Redprint-thinking – Focuses on stimulation of people, and implementing sophisticated HRM-instruments.

Greenprint-thinking – Focuses on ensuring that people are aware of new perspectives and personal shortcomings, while motivating them to see, learn, do new things, and create suitable shared learning experiences.

Whiteprint-thinking – Focuses on the natural flow of people's processes, interests and energies, and is concerned with the removal of blockades.

Each of these "colours" of thinking about change has their own merits. Depending on the organisational culture and architectural maturity in which an enterprise architect needs to operate, a different prevailing style will be needed.

The five roles from [132] can be mapped upon the competencies mentioned in Section 6.2. In most of these roles, communication, negotiation and sensitivity and empathy play a large role. Analytical skills and abstraction capacity are definitely needed for the modeller, but are also important to fulfil such a multidimensional role as enterprise architect. Using the competencies of enterprise architects as discussed in the previous Section, these roles can be made more specific as shown in Table 6.2. Note that we have treated the roles as "extremes" or "caricatures" when mapping the competencies. For example, to be a leader, an architect will also need some abstraction capacity. Nevertheless, the ability to abstract is really the core of their role as modeller. Conversely, when modelling, an architect also needs to be able to listen, which is a key trait for the communicator role.

In Table 6.2, the *change agent* role has been refined to include the colours of thinking about change discussed in [29]. In this table, we can see that the first four roles have many competencies in common, while the modeller is a complete different role.

Combining Table 6.1 with Table 6.2, results in Table 6.3. When examining this table, it is most striking to see that responsibilities and roles are not aligned to each other. Some responsibilities are attached to no role at all, while others are a combination of all roles. This really calls for future research. We have made some statements based upon this table, which of course need further research:

- No justice is done to the responsibilities involved in the maintanance of architectures. At the moment, only the modeller and blue change agent role are important for these.
- The communicator role seems less necessary than expected.

6.5 Enterprise architecture teams

Since enterprise architects are likely to operate in teams, it is not necessary to find a single person who fulfils all competencies. To combine a team of architects it is not

Roles / Competencies	Communicator	Leader	Manager	Modeller	Change Agent				
					Yellow Change Agent	Blue Change Agent	Red Change Agent	Green Change Agent	White Change Agent
Abstraction Capacity				x					
Accurateness			x	x		x			
Analytical Skills				x		x			
Authenticity		x							
Consulting	x	x							
Creativity		x		x				x	
Decisiveness		x	x			x	x		
Dedication		x		x		x	x		
Didactical Skills	x	x	x					x	
Diplomacy			x		x				
Facilitation skills	x		x						
Flexibility		x			x		x	x	x
Independency		x	x		x	x			x
Initiative		x	x						
Integrity		x	x				x		
Leadership		x	x		x				
Listening	x			x				x	
Loyalty									
Negotiation	x		x		x				
Openness									
Opinion Forming		x							x
Organisational Awareness			x		x	x	x	x	x
Persistence		x	x		x				
Persuasiveness	x	x			x		x		
Plan And Organize			x			x			
Result Driven			x						
Self Development		x						x	
Self-confidence		x			x			x	x
Sensitivity And Empathy		x	x				x	x	x
Stability		x	x		x				
Teamwork	x	x	x				x		
Verbal Communication	x	x	x	x	x	x	x	x	x
Visualisation skills	x			x					
Working Systematically			x	x		x			
Written Communication	x	x	x	x		x			

Table 6.2 Mapping competencies to roles and change colours

only necessary to find a good coverage of the competencies defined in Section 6.2, but also to ensure the group of selected architects indeed operates as a team. It is therefore also relevant to consider models for the abilities of people to work in teams. In [16] a number of roles of members in teams are identified:

Implementer – Well-organised and predictable. Takes basic ideas and makes them work in practice. Can be slow.

Shaper – Lots of energy and action, challenging others to move forwards. Can be insensitive.

Role / Architecture process (Ch. 5)	Communicator	Leader	Manager	Modeller	Change Agent: Yellow	Blue	Red	Green	White
Create									
Understand purpose and context			x	x		x	x		x
Determine deliverables	x	x	x	x		x	x		
Monitor context and stakeholders	x	x	x	x	x	x	x	x	x
Create shared conceptualisation	x	x	x	x	x	x	x	x	x
Design creation process									
Determine impacts									
Communicate	x	x	x		x			x	x
Apply									
Inform									x
Support decision-making	x	x	x	x	x	x	x	x	x
Ensure compliance	x	x	x	x	x	x	x		x
Make results available				x		x			
(Re-)communicate	x			x				x	
Maintain									
Monitor context & stakeholders						x			
Assess drivers for change						x			
Update & Communicate				x		x			
Organise									
Organise team					x		x		
Select framework, tools & tricks									
Communicate about EA	x	x	x		x			x	
Embed EA in Governance	x		x		x				
Monitor maturity									
Manage quality			x		x	x	x		x
Establish leadership	x	x	x		x		x	x	x
Innovate									

Table 6.3 Relating process and responsibilities to roles

Completer/Finisher – Reliably sees things through to the end, ironing out the wrinkles and ensuring everything works well. Can worry too much and not trust others.

Plant – Solves difficult problems with original and creative ideas. Can be poor communicator and may ignore the details.

Monitor/Evaluator – Sees the big picture. Thinks carefully and accurately about things. May lack energy or ability to inspire others.

Specialist – Has expert knowledge/skills in key areas and will solve many problems here. Can be disinterested in all other areas.

Coordinator – Respected leader who helps everyone focus on their task. Can be seen as excessively controlling.

Team worker – Cares for individuals and the team. Good listener and works to resolve social problems. Can have problems making difficult decisions.

Resource/investigator – Explores new ideas and possibilities with energy and with others. Good networker. Can be too optimistic and lose energy after the initial flush.

Within a team of enterprise architects there should be a balance between each of these roles. When considering the responsibilities identified in the previous Chapter, one can identify shifts in the priority that should be given to each of the involvement roles. We have made an attempt to achieve a mapping between the team involvement roles and the responsibilities of an enterprise architect (team), by comparing the competencies attached to a team role [16] with the competencies from Table 6.1. In creating the table, all role/responsibility combinations were selected were the team role had at least 60% of their underlying competencies in common with the competencies required by the responsibility. The result of this are shown in Table 6.4.

Architecture process \ Belbin-role	Completer	Coordinator	Implementer	Monitor	Plant	Resource investigator	Shaper	Teamworker
Create								
Understand purpose and context				x				
Determine deliverables	x					x	x	
Monitor context and stakeholders	x	x	x				x	
Create shared conceptualisation	x	x		x	x	x	x	x
Design creation process					x			
Determine impacts	x		x	x				
Communicate						x		x
Apply								
Inform								
Support decision-making	x	x	x	x	x	x		x
Ensure compliance	x	x	x	x				x
Make results available	x		x			x		
(Re-)communicate						x		x
Maintain								
Monitor context & stakeholders			x			x		
Assess drivers for change	x		x					
Update & Communicate	x		x					
Organise								
Organise team		x						
Select framework, tools & tricks								
Communicate about EA		x				x		x
Embed EA in Governance				x	x	x	x	
Monitor maturity	x		x					
Manage quality	x		x					
Establish leadership		x						x
Innovate								

Table 6.4 Belbin roles and the architecture process

The specialist is left out of scope for the comparison, because this is the person who is needed for expert roles, and less for his personal competencies. While

all roles are assigned to at least one responsibility, there are many responsibilities which are assigned to more than one role. Therefore, there seems to be no direct link between the roles and the responsibilities. An enterprise architect seems to be able to fulfill multiple roles for executing one responsibility. It is also striking to see that not all responsibilities are mapped to these roles. The "Inform" responsibility somehow is not mapped to Belbin-roles.

6.6 Professional development

Enterprise architecture is a field which is still developing. Is it already a profession? Or what should be done to become one? Finn defined six characteristics for a profession, namely [40]:

1. An intellectual technique.
2. An application of that technique to the practical affairs of man.
3. A period of long training necessary before entering into the profession.
4. An association of members of the profession into a closely-knit group with a high quality of communication between members.
5. A series of standards and a statement of ethics which is enforced.
6. An organised body of intellectual theory constantly expanding by research.

For each characteristic we will evaluate the state of enterprise architecture:

An intellectual technique – To be able to create a shared conceptualisation and to design the architecture, the enterprise architect needs to think reflectively, as well as to able to analyse and visualise results. Enterprise architects do own an intellectual technique to execute these tasks.

Practical application of that technique – Enterprise architectures are implemented in many enterprises.

Long period of training – A good enterprise architect needs a wide repertoire of professional and personal competencies.
Experience is one way of acquiring these competencies, while education is another. There are some Universities who offer programs for aspiring enterprise architects. Besides professional competencies, also the personal competencies are addressed in workshops. In addition to Universities, there are several training institutes who provide architectural courses with or without certification possibilities. For example:

- The Federal Enterprise Architect Certification Institute (FEACI) educates and certifies enterprise architects in terms of three distinct certification programmes, which are based on the Zachman Framework. These programmes are aimed at the professional competences of the architect.
- The Open Group has its own certification programme, referred to as ITAC (IT Architect certification programme), which is an independent and industry wide standard for IT Architects. At present this certification has no specific

requirements for enterprise architecture, but this is expected to be remedied in the near future. This certification is aimed at both professional as personal competencies.
- ERIA, the European Register for Information Architects, has its own certification programme called RACK (Regulation, Affection, Cognition and Knowledge). Besides taking the accrued experience into consideration, the ERIA certification program also uses assessments of both professional as personal competencies.
- Some large IT service companies have their own certification programmes, including Capgemini, HP and IBM.

While there are several certification programmes, all having their merits, there is not yet a widely accepted standard certification program. This makes certification and professional development of architects still rather fragmented.

Association and communication between members – There are some professional associations in the field of architecture, such as the Open Group, and the Netherlands Architecture Forum (NAF). However both recognize the field of enterprise architecture, they do not distinguish between types of architects yet.

Code of ethics and standards – There is not yet a code of ethics for enterprise architects. ACM has defined a code of ethics for business and IT professionals, and there are also associations within business professions who have these kinds of codes (such as doctors, lawyers, stockbrokers, consultants). The enforcement of some of these codes is very powerful, while for IT and consultancy codes, there are not yet many options for enforcement.

Standardisation is already a bit further in its development. There are architecture languages, styles and frameworks (see Chapter 4). Focussed on enterprise architecture specific, ArchiMate provides a powerful language, which is already embedded in some architecture tools.

Intellectual theory and research – Measuring enterprise architecture to this criterion, it can be concluded that the intellectual theory is still lacking. There are many practical implications, but often these are not underpinned on scientific literature.

In conclusion we can say that enterprise architecture is far in its practical use, however, lacks scientific sources. Therefore, enterprise architecture is not yet a profession, but it is well on its way to become one.

6.7 Summary

In this Chapter we discussed the basic competencies which an enterprise architect is expected to have, and tied these to the personality types needed to meet the responsibilities of architects as discussed in the previous Chapter. Though this match provides insight into the responsibilities, roles and competencies of architects, further research is needed. The alignment between roles and responsibilities was not

what we had expected. Some responsibilities are attached to no role at all, while others are a combination of all roles. Since enterprise architects are likely to operate in teams we also discussed the competencies needed to effectively work in teams.

We also discussed professional development of the enterprise architect. While there are several certification programmes, all having their merits, there is not yet a widely accepted standard certification program. This makes certification and professional development of architects still rather fragmented.

6.8 Discussion statements

1. An enterprise architect is far more important than the resulting enterprise architecture.
2. The programme manager and enterprise architect should bond together (be partners in crime)
3. An enterprise is better of hiring an external architect than employing one's own.
4. Communication about architecture is a profession of its own. An enterprise architect therefore does not have to worry about it.
5. An enterprise architect is only needed in the case of large scale changes in an enterprise. In any other case, an enterprise architect is just a burden.

Chapter 7
Conclusion

7.1 Summary

In this book we have explored the key concepts of enterprise architecture. An enterprise is understood to at least comprise of business, human and technological aspects. To be more precise, we defined an enterprise as *a goal oriented cooperative to be implemented by people and means*. In creating, evolving and/or transforming enterprises, several challenges come to the fore on how to govern such changes. Given these challenges, we stressed the important role of stakeholders, their stakes and concerns, as well as their needs with regard to an enterprise transformation. We also identified the notion of social complexity as a function of the number of stakeholders involved, variety of concerns, and the diversity in their backgrounds and abilities.

The increasing complexity of the issues facing the management of enterprises, as well as the growing diversity and heterogeneity of the concerns and stakes of the stakeholders involved, renders pre-existing approaches less adequate. We have identified enterprise architecture to be a means for enabling informed decision making on these changes, as well as ensuring compliance to these decisions made. As a next step, we discussed seven key applications for enterprise architecture: *situation description*, *strategic direction*, *gap analysis*, *tactical planning*, *operational planning*, *selection of partial solutions* and *solution architecture*, enabling informed governance.

Based on the enterprise architecture as a means for informed decision-making, we defined enterprise architecture to be:*A coherent set of descriptions, covering a regulations-oriented, design-oriented and patterns-oriented perspective on an enterprise, which provides indicators and controls that enable the informed governance of the enterprise's evolution and success*. The key concepts of enterprise architecture were identified as being: stakeholders, concerns, principles, models, views and frameworks. Using the discussion of the key concepts as a background, we showed the potential benefits of enterprise architecture, and the potential value

of architectural descriptions. We also stressed the fact that enterprise architecture is not the right means to be applied to every situation.

We have explored the results that may be yielded from enterprise architecting efforts, covering final deliverables, intermediate results and intangible results. We also explicitly visited the issue of the quality of the results. Depending on the intended use, different quality requirements should be met. We identified four key types of usage of results: specifying, deciding, informing, and contracting, also covering the justification of decisions made. This intended use has a great impact on architecture results needed. Based on the usage requirements of the involved stakeholders and their concerns, the intended result can be designed in terms of its subject and form.

Enterprise architecting involves a number of core processes: create, apply and maintain. We have shown that a good enterprise architecture process depends on a number of situational factors; therefore a one-size-fits-all approach does not exist. It is important to keep in mind that any enterprise architecture is a means to an end. It should deliver value by answering questions of stakeholders. It is therefore essential that the enterprise architecture processes do not solely focus on the delivering the enterprise architecture products, but start from the relevant stakeholders, and understanding their concerns, objectives and stated or implied requirements. Next to this criterion of effectiveness, the architecture process should be efficient, i.e., it should only develop those results (end results, intermediate results and intangible results) that are necessary to address the concerns of the stakeholders. At the same time the focus in the process should not be on the architecture results, but on the outcomes. We found that common descriptions of the architecture process currently tend to emphasise the creation (order of working and products) and sometimes the application of architecture results. In our view the application and maintenance of the enterprise architecture is equally important as the creation. In addition to activities, in which the process of architecting is *act*ed out, we also stressed the importance of *plan*ning, *learn*ing and *organis*ing activities. Activities involved in enterprise architecting should be scrutinized on their efficiency and effectiveness, and where possible, lessons learned should be recorded and taken into account in future situations. Combining the *act*ing, *plan*ning, and *learn*ing activities leads to a *plan-act-learn* cycle. In order to get this *plan-act-learn* cycle operational, and keep it operational, an explicit architecture function must be implemented in the enterprise (*organise*).

Frameworks and architecture process patterns, sometimes emerging into architecture schools are useful, but care needs to be taken when selecting them for use. This is a situational choice. Currently, few scientific publications exist that describe success factors for enterprise architecture processes. Most of the approaches and architecture maturity models are bundled as best practices of experts. Still, an architecture maturity model can help to cope with architecture maturity and increasing the maturity. The higher level of architecture maturity, the less focus will be on designing and implementing the architecture processes, but more on optimizing those processes. We have presented insight into the responsibilities, roles and competencies of architects, but concluded that further research is needed: the alignment between roles and responsibilities are not entirely consistent. We also discussed professional development of the enterprise architect. While there are several certifica-

tion programmes, all having their merits, there is not yet a widely accepted standard certification program. This makes certification and professional development of architects still rather fragmented.

7.2 Open issues

In this book we have explored the concepts of enterprise architecture, aiming to take a more fundamental view on enterprise architecture. In doing so, we have uncovered a number of challenges and open issues for the field of enterprise architecture that need further elaboration. Some challenges and issues need further investigation and scientific research, enabling the field of enterprise architecture to further mature. We realise that the field is (and needs to be) further developed in a collaboration between practitioners and the academic world.

Given the needs for enterprise architecture as discussed in Chapter 2, and the discussions provided in this book, one can conclude that our field is not yet mature, in the sense that as a profession we have to mature in helping organisations to solve their architectural problems in a *predictable and reproducible* way. To remedy this, several aspects of our field need further elaboration.

The aim of this Section is to identify and discuss some of the open issues. We will therefore provide an exposé of some key challenges for the field of enterprise architecting. The issues listed in this Section are an integration of three sources:

1. An innovation session, involving a mix of senior architects from Capgemini and researchers from Radboud University Nijmegen, involved in enterprise architecture practices, research and lecturing;
2. A survey of topics for potential master thesis projects. This survey was compiled by lecturers involved in the Master course on enterprise architecture taught in tandem by Capgemini and the Radboud University Nijmegen;
3. While writing this book, several additional challenges came to the fore.

The resulting set of issues and challenges have been grouped into: *the need*, *the results*, *the process* and *the architect*.

7.2.1 The need for enterprise architecture

As mentioned in Chapter 2, few empirical evidence exists on the value of enterprise architecture. A first important challenge is therefore:

The value of enterprise architecture – What is the value proposition of enterprise architecture? Does it really deliver the value promised? How to measure the value during and after the intended transformation? How should a business case for an enterprise architecture effort look like?

The hypothesis put forward in this book, is that enterprise architecture should fill the gap between strategy and design. This means it has two borders which may be disputed, leading to two important challenges:

Borderline between strategy and enterprise architecture – How can this borderline be defined clearly? Are strategy and enterprise architecture two sides of the same coin? How can strategy benefit from enterprise architecture? This question is not likely to lead to a one-size fits-all answer. But what are the mechanisms at work? How to deal with this in practical situations? How to maintain the link between strategy and architecture? How should strategists and enterprise architects best work together? If we regard architecture as an important means to manage risks, then this link should be addressed.

Borderline between enterprise architecture and design – This borderline is a critical one, as moving beyond it during architecting may unnecessarily lengthen the duration of architecture projects, threatening the original goal of being a steering instrument for change. Likewise, moving too soon from architecting into engineering/design might lead to the use of suboptimal solutions. One would expect this borderline to be situational. But to what extent? Where is the borderline? When is an enterprise architecture good enough? When to indeed move from architecting to engineering/designing? How to maintain the link between architecture and design? Will the border between architecture evolve (in a specific organisational context) over time? Is the borderline branch/domain/aspect specific?

7.2.2 The results of enterprise architecting

7.2.2.1 Increasing the value of deliverables

During enterprise architecting several architectural descriptions are created, such as design principles, models, views, etc. These descriptions have a potential value in terms of the insight they provide, their ability to steer/guide further developments, etc. Challenges remain on how to increase this potential value:

Selection of solution directions – One of the possible uses (and thus value) of architectural descriptions is the well-underpinned selecting between different alternatives. This promise, however, requires analytical models to indeed conduct architecture level analysis concerning different alternatives. In the ArchiMate project [78] some work was already performed on these issues, as well as work reported in e.g. [81, 96] on the selection of different organisational designs. Much work remains to be done, however, to be able to reliably predict performance, agility, robustness, etcetera, based on architectural descriptions.

Re-use of models – Enterprise architecting is a knowledge intensive activity. Hence it is sensible to look at the re-use of intermediate products. For example, branch and/or domain specific reference architectures/models, process specific refer-

ence models, solution strategy specific designs/models, etc. So the question arises, what are effective domains to invest in when aiming to achieve re-use of knowledge? What are these domains within our field, and how should we identify them? How to organise and maintain the resulting reference models? What is the benefit of a reference architecture? To what level of granularity is the reference architecture still generic?

Protection of modelling efforts – Architecture models are represented in some modelling/specification language. Such languages are supported by tools that allow for storage and manipulation of these models. When such models need to be exchanged between tools, teams and organisations, these languages need to be standardised. Is there indeed a need for interchange standards between architectural descriptions? Is there a need for something such as UML for enterprise architecting? What are the requirements on such standards? In the Netherlands ArchiMate [78] already seems to take this role. Does it meet the challenge? What role can be played by XMI, and OMG's MOF?

Putting principles to use – We consider principles to be the cornerstone of the regulation-oriented perspective on architecture. Before we can really turn principles into an effective regulative means, several questions need to be answered first: What is a principle? What is the added value of principles? How to best formulate principles? How to enforce them and/or use them to guide designers? How to make principles live up to their promise of being a steering instrument? How do they indeed impact on design decisions for enterprises as well as their IT systems? What is the cost of formulating and deploying a principle in relation to its benefit? How many principles are reasonable? For some initial work on these issues, see [23, 27, 98].

7.2.2.2 The creation and use of deliverables

The actual creation and implementation of enterprise architectures also poses several challenges:

Standard deliverables – Is it possible to define a standard set of architectural deliverables that need to be produced for specific classes of engagements? Is there a relationship between specific engagements and their situational context, and the selection of methods and techniques used to denote architectural descriptions?

Aptness of techniques and viewpoints – During an enterprise architecting process, several modelling techniques and viewpoints will be used. Which techniques and viewpoints should be used, for which audience, and for which purpose? In the ArchiMate project some initial theoretical results on these questions were reported [78, 108]. However, much work does indeed remain.

Understanding architectures – Architectures use terminology from the application domain. In order for an architecture to be communicated, the terminology used should be well defined (at least the core terminology), especially when the communication needs to span larger groups of people and/or when it needs to

bridge across longer periods of time. This essentially requires the creation of an explicit domain model/ontology [19]. Even more, enterprise models, be it at an architecture level or an engineering level, essentially involve concepts and their relations. Understanding these models therefore requires a proper understanding of these concepts and their relations [106, 109].

Standard description languages – The field of enterprise architecting certainly can do with more standardisation of terminology, and unifying on some core terminology is indeed desirable. Especially if this would lead to a notation with a unified look-and-feel, and a unification of the terminology of the underlying modelling concepts. In doing so, it would also be wise to extend/relate this terminology with concepts from business-oriented frameworks [7, 84].

Creation and selection of frameworks – Considering the discussion on architecture frameworks as provided in Chapter 4, it should be clear that there is no one-size-fits-all architecture framework. However, more research is needed to provide criteria supporting the selection/creation of an architecture framework in a specific situation. Work reported in e.g. [49, 56, 67, 149, 154] may provide a starting point.

7.2.2.3 Architectural content

Several classes of design decisions and stakeholder concerns also lead to challenges, calling for a methodological approach:

Deriving services – What would be a consistent way of deriving application services from information services, and in their turn from business services? Having a consistent way of doing this is especially important when applying the service oriented architecture style. A language such as ArchiMate does indeed allow one to express such a "chain". However, there is no methodological approach of deriving such a chain.

Specificity versus generality in defining services – Re-usability of services versus specificity for usage. When defining services, one can do so from the perspective of its potential re-use (i.e. being applicable/useable in a variety of situations), but also from the perspective of being as suitable/apt as possible for a specific usage goal. These to perspectives are likely to contradict each other. How to deal with this? How to strike a balance? Is this only a cost/benefit trade-off? Well-founded insight is needed here!

Change resiliency of application components – How (if at all) can application components (granularity and structure) be derived from business concepts (functions, goals, actors, transactions, etcetera) in a way which is resilient to organisational change, including organisational split-ups.

Risk management – Organisations are confronted with several risks. Enterprise architectures should therefore also be able to provide insight into how risks are dealt with. Either by preventing/reducing the probability of the risk occurring in the first place, or by reducing the impact if the risk does occur. These risks could pertain to many aspects, such as: operational risks, financial risks as well as se-

curity risks. It is likely that per class of risks, libraries of mechanisms/patterns can be compiled aimed at risk prevention or impact reduction. What are these mechanisms/patterns? Where are they applicable and/or effective?

Safety aspects – When considering an enterprise as a system, then there are three major areas from which risks may arise:

Systemic risks – there may be issues/flaws in the design of the system that cause it to function different from what was intended. These risks are typically covered by the governance aspect of an architecture framework such as IAF [30].

Inbound risks – actors involved in the execution of the processes in the enterprise, or actors in the environment of the enterprise, may deliberately or accidentally attack the stability/integrity/trustworthiness of the enterprise. These risks are typically covered by the security aspect of an architecture framework such as IAF.

Outbound risks – in executing the processes of the enterprise, these processes may lead to a security threat to other actors (inside or outside the enterprise). These latter class of risks are not taken into account in most architecture frameworks, and could/should lead to a *safety aspect* focussing on the potential (undesired) impact of the enterprise on its societal, business and physical environment.

7.2.3 The process of enterprise architecture

It is our observation that quite a lot of research has been done into architecture results, but hardly any into the architecture process.

7.2.3.1 Architecting from a cost/benefit perspective

Enterprise architecting involves effort. In other words, investments are made into the formulation of design principles, models, views, the implementation of architectures, etcetera. Some general questions to this point are: do these investments lead to a return on investment? And if so, what are they? How can we measure the costs of enterprise architecting efforts? How to guard these costs in relation to the potential return on investment?The question of a return on investment indeed is a much heard comment on enterprise architecting, and requires further investigation. Even more, when it becomes clearer what the return on investment is of several architecting activities, one is also able to better judge which activities to undertake in any given situation. This would enable a rationalisation of architecting activities in terms of their cost/benefit ratio. Two examples of situations which could benefit from a clearer understanding of the cost/benefit ratio are:

When to stop architecting? – In any given practical architecting situation, it is difficult to assess when enough is enough. In other words, it is hard to define a clear stop criterion to make a clear decision on when to stop detailing an architecture. When we would have a better understanding of value/cost/benefit ratios, it would be easier to define such a stopping criterion.

Just-in-time formality! – To operationalise deliverables of enterprise architecture towards (model-based) system development, the results need to have some level of formality. However, at the start of an enterprise architecture process it is too early to call for formalized results. So, where do we strike a balance? The benefit of producing formalized results also depends on the level of maturity of the architecture (and system development) process. If formal results are not utilized, the investment in their formalisation is in vain.

7.2.3.2 Shaping the process

The actual creation and implementation of enterprise architectures also poses several challenges:

Understanding and rationalising the process – What are possible strategies, processes and roadmaps to "do enterprise architecture"? What are the situational factors that influence the choice (and success) of these strategies? What are heuristics that would help in making the selection?

Predictable process – How to estimate the duration of an architecture creation and/or change project? How to make these processes more predictable? How to increase their speed? What is the frequency of these processes? Does this depend on a specific kind of organisation?

Lessons from quality management – How can (the maturity) of enterprise architecture and the drafted notion of *plan/act/learn* and *organise* benefit from generic quality frameworks such as EFQM [38] and Six Sigma [110]?

Success factors of enterprise architecting – Currently not much scientific results seem to be available describing the success factors for enterprise architecture processes. What are success factors? How well do projects score? Is it possible to define a generic set of critical success factors for all enterprise architectures?

Coping with power structures – Enterprise architecting processes – during creation, application and maintenance – have to deal/co-exist with social complexity and a pre-existing power structure in an organisation. How to deal with these? How to arrive at a truly shared conceptualisation in such situations? How to ensure that it consequently indeed implemented, while not suffering from "erosion" due to power games? How to apply architecture as an objective means to break up power games, and rather strive towards shared goals, which eventually could help to break down ineffective governance?

7.2.3.3 Architecture schools and styles

In the field of enterprise architecture, several schools and styles have come into existence. This calls for some comparative research, comparing between the different schools and styles.

Architecture schools and strategy schools – Analogous to architecture, where several schools exist, several schools exist in strategy formulation as well [85]. Due to the strong ties between strategy and enterprise architecture, it is only natural to wonder about the pitfalls and communalities between architecture and strategy schools. What are the underlying assumptions and situational dependencies of these schools? What can enterprise architecture schools learn from strategy schools?

Which schools and styles exist? – What are the contemporary architecture styles and schools? What is their underlying paradigm? What are their advantages and disadvantages?

When to use; and when not to use? – Which architecture school and style is applicable to a specific situation? Each architecture school needs instructions to assess its applicability in a specific situation. Furthermore, insight is needed into questions such as: What are known side-effects of applying an architecture school? What are the attention points and critical success factors?

Standard packages – Not much research is available on the impact of standard packages or product suites on the results and process of enterprise architecting. How to best align enterprise architecture and the high-level solution design, mandated by specific packages, from both a process and a content perspective? What enterprise architecture results are useful, when implementing a package? What is the added value of an enterprise architecture, if already is known that the chosen software package will be leading, also for organisational and process implementation?

7.2.4 The enterprise architect

We believe that the work on an architect's competencies as reported in Chapter 6, needs much more elaboration. Work has indeed been done, but much more rigorous work is needed still.

Improved terminological framework – Terminology such as skill, capability, competence, competency, personality type, role, etcetera, have different meanings in different sources. Before a further elaboration of the results reported in Chapter 6 is conducted, this terminological framework should be tightened up and made more explicit.

Cleaning up of the competencies list – The competencies listed in Chapter 6 need further scrutiny. The current list is based on initial work conducted within the Netherlands Architecture Forum [131, 145]. The list of competencies, however,

should be defined more precisely. Even more the orthogonality of the definitions should be improved. Some of the listed competencies seem to overlap, making empirical observations harder.

Further clarification of the responsibilities of architects – By defining the architecting process as revolving around create, apply, maintain and organise, it becomes possible to more explicitly assess in practice what these responsibilities entail. Further refinement of these responsibilities is therefore needed.

Alignment between roles and responsibilities – The alignment between roles and responsibilities was not what we had expected. Some responsibilities are attached to no role at all, while others are a combination of all roles. This needs some further investigation.

Linkage to education and training – Once the responsibilities and competencies have been determined, one can look into training and teaching programs. What can be taught by formal education at an academic level? What should be based on experience? What requires soft-skills training?

Code of ethics – Given a well understood list of responsibilities of an architect, it can also be identified what code of ethics and architect should abide by.

7.3 Further books needed in the Master of Enterprise Architecture program

As mentioned in the introductory Chapter, this book is positioned as a first in a series of books needed to underpin *Master of Enterprise Architecture* program with textbooks combining a sound theoretical base with practical insights. The program, and associated series of textbooks, is targeted both at university students, as well as practitioners with a keen interest in gaining a thorough understanding of these fields. Based on the curriculum, future books are expected to *at least* deal with:

- Architecture principles
- System theory for enterprise architects
- Business architecture
- Information architecture
- Application architecture

References

1. Oxford Dictionary of English, new ed edn. Oxford University Press, Oxford, United Kingdom, EU (2005), ISBN-13: 9780198610571
2. Architecture curriculum. Tech. rep., Capgemini (2007). URL http://academy.capgemini.com
3. Mergers and Acquisitions; Dangerous liaisons – the integration game. Research and opinions; executive summary, Hay Group (2007). URL http://www.haygroup.com
4. Aalst, W.v.d., Hofstede, A.t.: YAWL: yet another workflow language. Information Systems **30**(4), 245–275 (2005)
5. Alexander, C.: The Timeless Way of Building. Oxford University Press, Oxford, United Kingdom, EU (1979), ISBN-10: 0195024028
6. Amdahl, G., Blaauw, G., Brooks Jr, F.: Architecture of the ibm system/360. IBM Journal of Research and Development **8**(2) (1964)
7. Amelsvoort, P.v.: De moderne sociotechnische benadering – Een overzicht van de sociotechnische theorie. ST-Groep, Vlijmen, The Netherlands, EU (1999), ISBN-10: 9080138568. In Dutch
8. Apostel, L.: Towards the formal study of models in the non-formal sciences. Synthese (12), 125–161 (1960)
9. Arnold, B., Op 't Land, M.: An architectural approach to the implementation of shared service centers. In: D. Rijsenbrij (ed.) Proceedings of the Fourth National Architecture Congres (LAC2002) (2002). URL http://www.lac2002.nl
10. Arnold, B., Op 't Land, M., Dietz, J.: Effects of an architectural approach to the implementation of shared service centers. In: Second International Workshop on Enterprise, Applications and Services in the Finance Industry (FinanceCom05), Regensburg, Germany, EU (2005). URL http://www.iw.uni-karlsruhe.de/financecom05
11. Ashby, W.: An Introduction to Cybernetics. Chapman & Hall, London, United Kingdom, EU (1956), ISBN-10: 0412056704
12. Avison, D.: Information Systems Development: Methodologies, Techniques and Tools, 2nd edn. McGraw–Hill, New York, New York, USA (1995), ISBN-10: 0077092333
13. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison Wesley, Reading, Massachusetts, USA (1998), ISBN-10: 0201199300
14. Bean, S.: The elusive enterprise architect. IT Adviser (2006). URL http://www.nccmembership.co.uk
15. Beer, S.: Diagnosing the System for Organizations. Wiley, New York, New York, USA (1985)
16. Belbin, R.: Team Roles at Work. Butterworth Heinemann (1993), ISBN-10: 0750626755
17. Bergenhenegouwen, G., Mooijman, E., Tillema, H.: Strategic education and learning in organisations, 2nd edn. Kluwer, Deventer, The Netherlands, EU (1999). In Dutch

18. Blechar, M., Light, M.: Enterprise Information Architectures. Tech. Rep. R–450–131, GartnerGroup – ADM (1996). URL http://www.gartner.com
19. Bleeker, A., Proper, H., Hoppenbrouwers, S.: The Role of Concept Management in System Development – A practical and a theoretical perspective. In: J. Grabis, A. Persson, J. Stirna (eds.) Forum proceedings of the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004), Riga, Latvia, EU, pp. 73–82. Faculty of Computer Science and Information Technology, Riga, Latvia, EU (2004), ISBN-10: 998497670X
20. Boar, B.: Constructing Blueprints for Enterprise IT architectures. Wiley, New York, New York, USA (1999), ISBN-10: 0471296201
21. Boar, B.: Practical steps for aligning information technology with business strategies. Wiley, New York, New York, USA (1999), ISBN-10: 0471076376
22. Bommel, P.v., Buitenhuis, P., Hoppenbrouwers, S., Proper, H.: Architecture Principles – A Regulative Perspective on Enterprise Architecture. In: M. Reichert, S. Strecker, K. Turowski (eds.) Enterprise Modelling and Information Systems Architectures (EMISA2007), no. 119 in Lecture Notes in Informatics, pp. 47–60. Gesellschaft fur Informatik, Bonn, Germany, EU (2007)
23. Bommel, P.v., Hoppenbrouwers, S., Proper, H., Weide, T.v.d.: Giving meaning to enterprise architectures – architecture principles with orm and orc. In: R. Meersman, Z. Tari, P. Herrero (eds.) On the Move to Meaningful Internet Systems 2006: OTM Workshops – OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, GADA, MIOS+INTEROP, ORM, PhDS, SeBGIS, SWWS, and WOSE 2006, Lecture Notes in Computer Science. Springer, Berlin, Germany, EU, Montpellier, France, EU (2006)
24. Bommel, P.v., Hoppenbrouwers, S., Proper, H., Weide, T.v.d.: QoMo: A Modelling Process Quality Framework based on SEQUAL. In: H. Proper, T. Halpin, J. Krogstie (eds.) Proceedings of the 12th Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD'07), held in conjunctiun with the 19th Conference on Advanced Information Systems (CAiSE'07), Trondheim, Norway, pp. 118–127. Tapir Academic Press, Trondheim, Norway (2007), ISBN-10: 9788251922456
25. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modelling Language User Guide. Addison Wesley, Reading, Massachusetts, USA (1999), ISBN-10: 0201571684
26. Bredemeyer, D., Malan, R.: What It Takes to Be a Great Enterprise Architect. Enterprise Architecture - Cutter Consortium **7**(8) (2004)
27. Buitenhuis, P.: Fundamenten van het principle (Foundations of principles). Master's thesis, Institute for Computing and Information Sciences, Radboud University Nijmegen, Nijmegen, The Netherlands, EU (2007). In Dutch
28. Bunge, M.: A World of Systems, *Treatise on Basic Philosophy*, vol. 4. D. Reidel Publishing Company, Dordrecht, The Netherlands, EU (1979)
29. de Caluwé, L., Vermaak, H.: Learning to Change: A Guide for Organization Change Agents. Sage publications, London, United Kingdom, EU (2003), ISBN-10: 9014961587
30. Capgemini: Enterprise, Business and IT Architecture and the Integrated Architecture Framework. Service Oriented Architecture - The way we see it. Utrecht, The Netherlands, EU (2007)
31. Conklin, J.: Wicked Problems and Social Complexity. Tech. rep., CogNexus Institute, Edgewater, Maryland, USA (2006). URL http://cognexus.org
32. Daft, R.: Organization Theory and Design, 9th edn. South-Western College Pub, San Diego, California, USA (2006), ISBN-10: 0324405421
33. Department of Commerce, Government of the USA: Introduction - IT Architecture Capability Maturity Model. Government of the United States of America (2003). URL http://ocio.os.doc.gov/groups/public/@doc/@os/@ocio/@oitpp/documents/content/prod01_002340.pdf
34. Dietz, J.: Generic recurrent patterns in business processes. In: W.v.d. Aalst, A.t. Hofstede, M. Weske (eds.) Proceedings of the International Conference on Business Process Management (BPM2003), Eindhoven, The Netherlands, EU, *Lecture Notes in Computer Science*, vol. 2678, pp. 1018–1033. Springer-Verlag, Berlin, Germany, EU (2003), ISBN-10: 3540403183

35. Dietz, J.: Enterprise Ontology – Theory and Methodology. Springer, Berlin, Germany, EU (2006), ISBN-10: 9783540291695
36. Dietz, J., Go A. en Lee, C.: Enterprise Architecture in de praktijk - Het belang van awareness . Via Nova Architectura (2007). URL http://www.via-nova-architectura.org/. In Dutch
37. Dietz, J., Hoogervorst, J.: Enterprise Ontology and Enterprise Architecture – how to let them evolve into effective complementary notions. GEAO Journal of Enterprise Architecture **1** (2007)
38. EFQM: The efqm excellence model (2008). URL http://www.efqm.org
39. Eriksson, H.E., Penker, M.: Business Modeling with UML: Business Patterns at Work. Wiley, New York, New York, USA (1998)
40. Finn, J.: Professionalizing the audio-visual field. Educational Technology Research and Development **1**(1), 6–17 (1953), ISSN 1042-1629. DOI 10.1007/BF02713166
41. Foorthuis, R., Brinkkemper, S.: A framework for project architecture in the context of enterprise architecture. Via Nova Architectura (2007). URL http://www.via-nova-architectura.org/
42. Franckson, M., Verhoef, T. (eds.): Introduction to ISPL. Information Services Procurement Library. ten Hagen & Stam, Den Haag, The Netherlands, EU (1999), ISBN-10: 9076304858
43. Friedman, T.: The World is Flat: A Brief History of the Twenty-first Century. Farrar, Straus and Giroux, New York, New York, USA (2005), ISBN-10: 0374292884
44. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object–Oriented Software. Addison Wesley, Reading, Massachusetts, USA (1995)
45. Goedvolk, J., Bruin, H.d., Rijsenbrij, D.: Integrated architectural design of business and information systems. In: J. Bosch (ed.) Proceedings of the Second Nordic Workshop on Software Architecture (NOSA'99), *Research Report*, vol. 1999. University of Karlskrona/Ronneby, Ronneby, Sweden, EU (1999), ISSN: 1103-1581
46. Gordijn, J., Akkermans, H.: Value based requirements engineering: Exploring innovative e-commerce ideas. Requirements Engineering Journal **8**(2), 114–134 (2003). DOI 10.1007/s00766-003-0169-x
47. Government of the USA: Sarbanes-Oxley Act of 2002. H.R.3763 (2002)
48. Graaf, E.v.d.: Architectuurprincipes en clustercriteria voor de afbakening van out-sourcebare kavels. Master's thesis, Radboud University Nijmegen (2006). In Dutch
49. Greefhorst, D., Koning, H., Vliet, H.v.: Dimensies in architectuurbeschrijvingen. Informatie **45**(11), 22–27 (2003). In Dutch
50. Green, N., Bate, C.: Lost in Translation – A handbook for information systems in the 21st century. Evolved Technologist Press, New York, New York, USA (2007), ISBN-13: 9780978921842
51. Grembergen, W.v., Saull, R.: Aligning business and information technology through the balanced scorecard at a major canadian financial group: its status measured with an it bsc maturity model (2001). URL http://www.hicss.hawaii.edu/HICSS_34/PDFs/OSKBE03.pdf
52. Groote, G., Hugenholtz-Sasse, C., Slikker, P.: Projecten leiden: Methoden en technieken voor projectmatig werken. Het Spectrum, Utrecht, The Netherlands, EU (1995), ISBN-10: 9027497605. In Dutch
53. Hagel III, J., Armstrong, A.: Net Gain – Expanding markets through virtual communities. Harvard Business School Press, Boston, Massachusetts, USA (1997)
54. Hakes, C.: The EFQM Excellence Model to Assess Organizational Performance - A Management Guide. Van Haren Publishing (2007), ISBN-13: 978-9087530273
55. Henderson, J., Venkatraman, N.: Strategic alignment: Leveraging information technology for transforming organizations. IBM Systems Journal **32**(1), 4–16 (1993)
56. Heuvel, W.J.v.d., Proper, H.: De pragmatiek van Architectuur. Informatie **44**(11), 12–14 (2002). In Dutch
57. Hoppenbrouwers, S., Proper, H., Bleeker, A.: Modelleertalen als communicatiemiddel in architectuurprocessen. Informatie & Architectuur **1**(1), 4–6 (2005). In Dutch

58. Horan, T.: Digital Places – Building our city of bits. The Urban Land Institute (ULI), Washington DC, USA (2000), ISBN-10: 0874208459
59. Humphrey, W.: Managing the Software Process. The SEI Series in Software Engineering. Addison-Wesley Professional, Massachusetts, USA (1989), ISBN-13: 9780201180954
60. Recommended Practice for Architectural Description of Software Intensive Systems. Tech. Rep. IEEE P1471–2000, The Architecture Working Group of the Software Engineering Committee, Standards Department, IEEE, Piscataway, New Jersey, USA (2000), ISBN-10: 0738125180. URL `http://www.ieee.org`
61. IFIP-IFAC Task Force: GERAM: Generalised Enterprise Reference Architecture and Methodology (1999). Version 1.6.3, Published as Annex to ISO WD15704
62. Immink, J., Hendrickx, C.: Kiezen voor delen, succes verzekerd. Eye on Finance, Ernst & Young, Rotterdam, The Netherlands, EU (2002). In Dutch
63. Information processing systems – Concepts and Terminology for the Conceptual Schema and the Information Base (1987). URL `http://www.iso.org`. ISO/TR 9007:1987
64. ISO: Information technology – Open Distributed Processing – Reference model: Architecture (1996). URL `http://www.iso.org`. ISO/IEC 10746–3:1996(E)
65. ISO: Kwaliteit van softwareprodukten. ten Hagen & Stam, Den Haag, The Netherlands, EU (1996), ISBN-10: 9026724306. In Dutch
66. Software engineering – Product quality – Part 1: Quality model (2001). URL `http://www.iso.org`. ISO/IEC 9126–1:2001
67. Janssen, R., Proper, H., Bosma, H., Verhoef, D., Hoppenbrouwers, S.: Developing an Architecture Method Library. Tech. rep., Ordina Institute, Gouda, The Netherlands, EU (2001)
68. Johnson, G., Scholes, K., Whittington, R.: Exploring Corporate Strategy, seventh edn. Prentice Hall, Englewood Cliffs, New Jersey, USA (2005), ISBN-10: 0273687344
69. Jonkers, H., Lankhorst, M., Buuren, R.v., Hoppenbrouwers, S., Bonsangue, M., Torre, L.v.d.: Concepts for Modeling Enterprise Architectures. International Journal of Cooperative Information Systems **13**(3), 257–288 (2004)
70. Jonkers, H., Veldhuijzen van Zanten, G., Buuren, R.v., Arbab, F., Boer, F.d., Bonsangue, M., Bosma, H., Doest, H.t., Groenewegen, L., Guillen Scholten, J., Hoppenbrouwers, S., Iacob, M.E., Janssen, W., Lankhorst, M., Leeuwen, D.v., Proper, H., Stam, A., Torre, L.v.d.: Towards a Language for Coherent Enterprise Architecture Descriptions. In: M. Steen, B. Bryant (eds.) 7th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2003), Brisbane, Queensland, Australia, pp. 28–39. IEEE, Los Alamitos, California, USA (2003), ISBN-10: 0769519946
71. Kaplan, R., Norton, D.: The Strategy-Focused Organization: How Balanced Scorecard Companies Thrive in the New Business Environment. Harvard Business School Press, Boston, Massachusetts, USA (2000), ISBN: 1578512506
72. Kensing, F.: Towards Evaluation of Methods for Property Determination: A Framework and a Critique of the Yourdon–DeMarco Approach. In: T. Bemelmans (ed.) Beyond Productivity: Information Systems Development for Organizational Effectiveness, Amsterdam, The Netherlands, EU, pp. 325–338. North–Holland, Amsterdam, The Netherlands, EU (1984)
73. Kolfschoten, G., Vreede, G.J.d., Chakrapani, A., Koneri, P.: The Collaboration Engineering Approach for Designing Collaboration Processes. In: Proceedings of the First HICSS Symposium on Case and Field Studies of Collaboration, Poipu, Kauai, Hawaii (2006)
74. Krogstie, J.: A Semiotic Approach to Quality in Requirements Specifications. In: L. Kecheng, R. Clarke, P. Andersen, R. Stamper, E.S. Abou–Zeid (eds.) Proceedings of the IFIP TC8 / WG8.1 Working Conference on Organizational Semiotics: Evolving a Science of Information Systems, pp. 231–250. Kluwer, Deventer, The Netherlands, EU (2002), ISBN-10: 1402071892
75. Krogstie, J., H.D., J.: Quality of Interactive Models. In: M. Genero, G. F., W.J.v.d. Heuvel, J. Krogstie, K. Lyytinen, H. Mayr, J. Nelson, A. Olivé, M. Piattine, G. Poels, J. Roddick, K. Siau, M. Yoshikawa, E. Yu (eds.) 21st International Conference on Conceptual Modeling (ER 2002), *Lecture Notes in Computer Science*, vol. 2503, pp. 351–363. Springer, Berlin, Germany, EU (2002), ISBN-10: 3540442774. DOI 10.1007/b12013

76. Krogstie, J., Sølvberg, A.: Information Systems Engineering – Conceptual Modeling in a Quality Perspective. The Norwegian University of Science and Technology, Stockholm, Norway (2000)
77. Kruchten, P.: The 4+1 View Model of Architecture. IEEE Software **12**(6), 42–50 (1995)
78. Lankhorst, M., et al.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Berlin, Germany, EU (2005), ISBN-10: 3540243712
79. Leeuw, A.d.: Organisaties: Management, Analyse, Ontwikkeling en Verandering, een systeem visie. van Gorcum, Assen, The Netherlands, EU (1982). In Dutch
80. Leeuw, A.d., Volberda, H.: On the Concept of Flexibility: A Dual Control Perspective. Omega, International Journal of Management science **24**(2), 121–139 (1996)
81. M., O.: Applying Architecture and Ontology in the Splitting and Allying of Enterprises (2008)
82. Maier, M., Rechtin, R.: The Art of System Architecting, 2nd edn. CRC Press, Boca Raton, Florida, USA (2002), ISBN-10: 0849304407
83. Mathiassen, L.: Systemudvikling og Systemudviklings–Metode. Ph.D. thesis, Aarhus University, Aarhus, Denmark, EU (1981). In Danish
84. McDavid, D.: A standard for business architecture description. IBM Systems Journal **38**(1), 12–31 (1999)
85. Mintzberg, H., Ahlstrand, B., Lampel, J.: Strategy safari – A guided tour through the wilds of strategic management. The Free Press, New York, New York, USA (1998), ISBN-10: 0684847434
86. Mulder, J.: Rapid Enterprise Design (2007), ISBN-10: 9081048015
87. Mulder, L.: Pda comes to the aid of police officers in groningen (zakcomputer helpt agenten in groningen). Algemeen Dagblad (19th of April) (2007)
88. Mulholland, A., Thomas, C., Kurchina, P., Woods, D.: Mashup Corporations - The End of Business as Usual. Evolved Technologist Press, New York, New York, USA (2006), ISBN-13: 9780978921804
89. Nabukenya, J., Bommel, P.v., Proper, H.: Collaborative IT Policy-making as a means of achieving Business-IT Alignment. In: B. Pernici, J. Gulla (eds.) Proceedings of the Workshop on Business/IT Alignment and Interoperability (BUSITAL'07), held in conjunctiun with the 19th Conference on Advanced Information Systems (CAiSE'07), Trondheim, Norway, pp. 461–468. Tapir Academic Press, Trondheim, Norway (2007), ISBN-10: 9788251922456
90. NAF working group: Professionalisation of the Architect: Competenties van de ICT-Architect. Netherlands Architecture Forum (2006.)
91. Olle, T., Hagelstein, J., Macdonald, I., Rolland, C., Sol, H., Assche, F.v., Verrijn–Stuart, A.: Information Systems Methodologies: A Framework for Understanding. Addison Wesley, Reading, Massachusetts, USA (1988), ISBN-10: 0201544431
92. OMG: UML 2.0 Superstructure Specification – Final Adopted Specification. Tech. Rep. ptc/03–08–02 (2003). URL `http://www.omg.org`
93. Op 't Land, M.: Applying Architecture and Ontology to the Splitting and Allying of Enterprises: Problem Definition and Research Approach. In: R. Meersman, Z. Tari, P. Herrero (eds.) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops - OTM Confederated International Workshops and Posters, AWESOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Acadamy Doctoral Consortium, RDDS, SWWS, and SebGIS, Proceedings, Part II, Montpellier, France, EU, *Lecture Notes in Computer Science*, vol. 4278, pp. 1419–1428. Springer, Berlin, Germany, EU (2006)
94. Op 't Land, M.: Towards Evidence Based Splitting of Organizations. In: J. Ralyté, S. Brinkkemper, B. Henderson-Sellers (eds.) Proceedings of the IFIP TC8 / WG8.1 Working Conference on Situational Method Engineering: Fundamentals and Experiences (ME07), Geneva, Switzerland, *IFIP Series*, vol. 244, pp. 328–342. Springer, Berlin, Germany, EU (2007), ISBN-10: 9780387739465
95. Op 't Land, M., Arnold, B., Engels, A.: Fps: another way of looking at components and architecture in the financial world. In: Proceedings of the Second National Architecture Conference (LAC2000) (2000)

96. Op 't Land, M., Dietz, J.: Enterprise ontology based splitting and contracting of organizations. In: Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC'08), Fortaleza, Ceará, Brazil (2008)
97. Op 't Land, M., Middeljans, K., Buller, V.: Enterprise Ontology based Application Portfolio Rationalization at Rijkswaterstaat. In: The 4th Dutch Championship ICT Architecture (2007). URL http://www.nkictarchitectuur.nl/2007
98. Op 't Land, M., Proper, H.: Impact of Principles on Enterprise Engineering. In: H. Österle, J. Schelp, R. Winter (eds.) Proceedings of the 15th European Conference on Information Systems, pp. 1965–1976. University of St. Gallen, St. Gallen, Switzerland (2007)
99. Parker, M., Benson, R.: Enterprisewide Information Management: State–of–the–art Strategic Planning. Journal of Information Systems Management (Summer), 14–23 (1989)
100. Parnas, D.: Information Distribution Aspects of Design Methodology. In: C. Freiman, J. Griffith, J. Rosenfeld (eds.) Information Processing 71, Proceedings of IFIP Congress 71, Ljubljana, Slovenia, EU, vol. 1 - Founda. North–Holland, Amsterdam, The Netherlands, EU (1971), ISBN-10: 0720420636
101. Pohl, K.: The three dimensions of requirements engineering: a framework and its applications. Information Systems **19**(3), 243–258 (1994), ISSN: 03064379
102. Porter, M.: What is strategy. Harvard Business Review (November – December) (1996)
103. Price, C.: Business and the art of transformation. The McKinsey Quarterly (3) (2006)
104. Proper, H.: ISPL toegepast – Aanbestedingen van grootschalige migratieprojecten. ID Nieuws **3**, 4–8 (1999). In Dutch
105. Proper, H. (ed.): ISP for Large–scale Migrations. Information Services Procurement Library. ten Hagen & Stam, Den Haag, The Netherlands, EU (2001), ISBN-10: 9076304882
106. Proper, H., Bommel, P.v., Hoppenbrouwers, S., Weide, T.v.d.: A fundamental view on the act of modeling. In: J. Kizza, J. Aisbett, A. Vince, T. Wanyama (eds.) Advances in Systems Modelling and ICT Applications, *Special topics in computing and ICT research*, vol. 2. Fountain Publishers, Kampala, Uganda, Kampala, Uganda (2006)
107. Proper, H., Hoppenbrouwers, S., Veldhuijzen van Zanten, G.: Communication of Enterprise Architectures. In: Enterprise Architecture at Work: Modelling, Communication and Analysis [78], pp. 67–82, ISBN-10: 3540243712
108. Proper, H., Verrijn–Stuart, A., Hoppenbrouwers, S.: Towards Utility–based Selection of Architecture–Modelling Concepts. In: S. Hartmann, M. Stumptner (eds.) Proceedings of the Second Asia–Pacific Conference on Conceptual Modelling (APCCM2005), Newcastle, New South Wales, Australia, *Conferences in Research and Practice in Information Technology Series*, vol. 42, pp. 25–36. Australian Computer Society, Sydney, New South Wales, Australia (2005), ISBN-10: 1920682252
109. Proper, H., Weide, T.v.d.: Modelling as Selection of Interpretation. In: H. Mayr, H. Breu (eds.) Modellierung 2006, *Lecture Notes in Informatics*, vol. P82, pp. 223–232. Gesellschaft für Informatik, Bonn, Germany, EU (2006), ISBN-10: 3885791765
110. Pyzdek, T.: The six sigma handbook: The complete guide for greenbelts, blackbelts, and managers at all levels, revised and expanded edition (2003), ISBN-13: 9780071410151
111. Raadt, B.v.d.: Normalized Architecture Organization Maturity Index. Tech. rep., Capgemini (2006)
112. Raadt, B.v.d., Slot, R., Vliet, H.v.: Experience Report: Assessing a Global Financial Services Company on its Enterprise Architecture Effectiveness Using NAOMI pp. 218b–218b (2007), ISBN-10: 0769527558. DOI 10.1109/HICSS.2007.217
113. Rechtin, E.: Systems architecting: creating and building complex systems. Prentice–Hall, Englewood Cliffs, New Jersey, USA (1991), ISBN-10: 0138803455
114. Reijswoud, V.v., Dietz, J.: DEMO Modelling Handbook, vol. 1, 2nd edn. Delft University of Technology, Delft, The Netherlands, EU (1999)
115. Rijsenbrij, D.: Outsourcing zonder enterprise architectuur lijkt op autorijden zonder veiligheidsgordel. Tech. Rep. NIII–R0404, Nijmegen Institute for Information and Computing Sciences, University of Nijmegen, Nijmegen, The Netherlands, EU (2004)

116. Rijsenbrij, D., Schekkerman, J., Hendrickx, H.: Architectuur, besturingsinstrument voor adaptieve organisaties – De rol van architectuur in het besluitvormingsproces en de vormgeving van de informatievoorziening. Lemma, Utrecht, The Netherlands, EU (2002), ISBN-10: 9059310934. In Dutch
117. Rittel, H., Webber, M.: Dilemmas in a General Theory of Planning. Policy Sciences **4**, 155–169 (1973)
118. Ross, J., Weill, P., Robertson, D.: Enterprise architecture as strategy: creating a foundation for business execution. Harvard Business School Press, Boston, Massachusetts, USA (2006), ISBN-10: 1591398398
119. Sanden, W.v.d., Sturm, B.: Informatie–architectuur – de infrastructurele benadering. Panfox, Rosmalen, The Netherlands, EU (1997), ISBN-10: 9080127027. In Dutch
120. Schekkerman, J.: Enterprise architecture score card (2004)
121. Schelp, J., Stutz, M.: A Balanced Scorecard Approach to Measure the Value of Enterprise Architecture. In: M. Lankhorst, P. Johnson (eds.) Proceedings of the Second Workshop on Trends in Enterprise Architecture Research (TEAR 2007) (2007). URL `http://www.via-nova-architectura.org/`
122. Seligmann, P., Wijers, G., Sol, H.: Analyzing the Structure of I.S. Methodologies, an alternative approach (1989)
123. Shaw, M., Garlan, D.: Software Architecture: Perspectives on an Emerging Discipline. Prentice–Hall, Englewood Cliffs, New Jersey, USA (1996), ISBN-10: 0131829572
124. Sitter, L.d.: Synergetisch produceren; Human Resources Mobilisation in de produktie: een inleiding in structuurbouw. Van Gorcum, Assen, The Netherlands, EU (1998), ISBN-13: 9789023233657. In Dutch
125. Smit, R.: Manieren om strategie te verknallen. Financieel Dagblad (23 February, 2007) (2007). URL `http://www.fd.nl/`. In Dutch
126. Sol, H.: A Feature Analysis of Information Systems Design Methodologies: Methodological Considerations. In: T. Olle, H. Sol, C. Tully (eds.) Information Systems Design Methodologies: A Feature Analysis, Amsterdam, The Netherlands, EU, pp. 1–7. North–Holland/IFIP WG8.1, Amsterdam, The Netherlands, EU (1983), ISBN-10: 0444867058
127. Sol, H.: Information Systems Development: A Problem Solving Approach. In: W. Cotterman, J. Senn (eds.) Proceedings of 1988 INTEC Symposium Systems Analysis and Design: A Research Strategy. Georgia State University, Atlanta, Georgia (1988)
128. Sowa, J., Zachman, J.: Extending and formalizing the framework for information systems architecture. IBM Systems Journal **31**(3), 590–616 (1992)
129. Stachowiak, H.: Allgemeine Modelltheorie. Springer, Berlin, Germany, EU (1973), ISBN-10: 3211811060
130. Steghuis, C., Proper, H.: Competencies and responsibilities of Enterprise Architects – A jack-of-all-trades? In: J. Dietz, T. Albani, J. Barjis, P. Rittgen (eds.) Advances in Enterprise Engineering – Proceedings of the CIAO!-EOMAS'08 workshops, Lecture Notes in Business Information Processing. Springer, Berlin, Germany, EU (2008). To be published
131. Steghuis, C., Voermans, K., Wieringa, R.: Competencies of the ICT architect. Tech. rep., Netherlands Architecture Forum (2005)
132. Strano, C., Rehmani, Q.: The role of the enterprise architect. Information Systems and E-Business Management **5**(4), 379–396 (2007). DOI 10.1007/s10257-007-0053-1
133. Tak, T.v.d., Wijnen, G.: Programmamanagement – Sturen op samenhang, 2nd edn. Kluwer (2006), ISBN-13: 978901303628. In Dutch
134. Tapscott, D.: Digital Economy – Promise and peril in the age of networked intelligence. McGraw–Hill, New York, New York, USA (1996), ISBN-10: 0070633428
135. Tapscott, D., Caston, A.: Paradigm Shift – The New Promise of Information Technology. McGraw–Hill, New York, New York, USA (1993), ASIN 0070628572
136. The Open Group: A Pocket Guide to TOGAF Version 8.1.1 Enterprise Edition. The Open Group (2007). URL `http://www.opengroup.org/togaf`
137. Thomson, A., Martinet, A.: A Practical English Grammar, fourth edn. Oxford University Press, Oxford, United Kingdom, EU (1986), ISBN-10: 3810905798

138. TM Forum: Enhanced Telecom Operations Map (eTOM) – The Business Process Framework For The Information and Communications Services Industry. GB921v3.5. TM Forum, Morristown, New Jersey, USA (2003). URL http://www.tmforum.org
139. TOGAF – The Open Group Architectural Framework (2005). URL http://www.togaf.org
140. Treacy, M., Wiersema, F.: The Discipline of Market Leaders – Choose your customers, narrow your focus, dominate your market. Addison Wesley, Reading, Massachusetts, USA (1997), ISBN-10: 0201407191
141. Umar, A.: It infrastructure to enable next generation enterprises. Information Systems Frontiers **7**(3), 217–256 (2005)
142. USA Government: Clinger–Cohen; IT Management Reform Act (1996). URL http://www.cio.gov/Documents/it_management_reform_act_Feb_1996.html
143. Vennix, J.: Mental models and Computer models: design and evaluation of a computer–based learning environment for policy making. Ph.D. thesis, Proefschrift Katholieke Universiteit Brabant, University of Nijmegen (1990)
144. Vennix, J.: Group Model Building: facilitating team learning using systems dynamics. Wiley, New York, New York, USA (1996)
145. Voermans, K., Steghuis, C., Wieringa, R.: Architect roles and competencies – A questionnaire conducted during the Dutch Architectural Conference 2004. Tech. rep., Netherlands Architecture Forum (2005). In Dutch
146. Vreede, G.d.: Collaboration Engineering: Designing for Self–Directed Team Efforts. In: Proceedings of the Shaping the Future of IT (2004). Conference, Tucson, AZ, November 3–5
147. Wagter, R., Berg, M.v.d., Luijpers, J.: DYA: snelheid en samenhang in business en ICT architectuur. Tutein Nolthenius (2001), ISBN: 9072194624
148. Wagter, R., Berg, M.v.d., Luijpers, J., Steenbergen, M.v.: Dynamic Enterprise Architecture: How to Make It Work. Wiley, New York, New York, USA (2005), ISBN-10: 0471682721
149. Wagter, R., Nijkamp, G., Proper, H.: The Elements of the "GEA-Structure". White Paper GEA-2, Ordina, Utrecht, The Netherlands, EU (2007). In Dutch
150. Wagter, R., Witte, D., Proper, H.: The GEA architecture function: A strategic specialism. White Paper GEA-7, Ordina, Utrecht, The Netherlands, EU (2007). In Dutch
151. Walker, M.: A day in the life of an enterprise architect. Tech. rep., Microsoft corporation (2007). URL http://msdn2.microsoft.com/en-us/library/bb945098.aspx
152. Wijers, G., Heijes, H.: Automated Support of the Modelling Process: A view based on experiments with expert information engineers. In: B. Steinholz, A. Sølvberg, L. Bergman (eds.) Proceedings of the Second Nordic Conference CAiSE'90 on Advanced Information Systems Engineering, Stockholm, Sweden, EU, *Lecture Notes in Computer Science*, vol. 436, pp. 88–108. Springer, Berlin, Germany, EU (1990), ISBN-10: 3540526250
153. Wood–Harper, A., Antill, L., Avison, D.: Information Systems Definition: The Multiview Approach. Blackwell, Oxford, United Kingdom, EU (1985), ISBN-10: 0632012168
154. xAF working group: Extensible Architecture Framework version 1.1 (formal edition). Tech. rep. (2006). URL http://www.xaf.nl
155. Zachman, J.: A framework for information systems architecture. IBM Systems Journal **26**(3) (1987)
156. Zagotta, R., Robinson, D.: Keys to successful strategy execution: The most brilliant strategy ever devised won't get you anywhere if you can't execute it. Journal of Business Strategy, Emerald **23** (2002), ISSN: 02756668

About the authors

- Martin Op 't Land is Principal Consultant and Enterprise Architect at Capgemini and has over 16 years of experience in the field of enterprise architecture, mostly in the finance and public sectors. In addition, he is involved in research in the field of enterprise engineering and enterprise architecting. In line with his PhD-thesis 'Applying Architecture and Ontology to the Splitting and Allying of Enterprises', his current research focuses on methods for organisation splitting, organisation allying and post-merger integration.
- Erik Proper is Principal Consultant at Capgemini and Professor in Information Systems at the Radboud University Nijmegen. He has a mixed industrial and academic background. His interests lie mainly in the field of conceptual modelling, enterprise modelling, enterprise engineering and enterprise architecting. He was co-initiator of the ArchiMate project, and currently also serves on the board of the ArchiMate foundation as well as the Netherlands Architecture Forum (NAF).
- Maarten Waage is Enterprise Architect at Capgemini. He has over ten years experience in creating and applying enterprise architecture, mostly in the context of complex business/IT transformations of large organisations (both profit and not-for-profit).
- Jeroen Cloo is Senior Consultant and Enterprise Architect at SeederDeBoer. After obtaining a bachelor degree in Technology Management, he started as an consultant over ten years ago. In his career, he has been involved in multiple enterprise architecture engagements in different sectors. Jeroen obtained a master degree in Enterprise Architecture at the Radboud University Nijmegen.
- Claudia Steghuis is senior consultant at Capgemini with about two years of experience in enterprise architecture. Claudia obtained a master degree in Business information technology, discipline BIT Architecture at the University of Twente. Her MSc research focussed on service granularity within service oriented architecture. She was also involved in research conducted by the Netherlands Architecture Forum on the competencies of the architect.

Martin, Erik and Maarten are also involved as lecturers in the post academic master program on Enterprise Architecture offered by the Radboud University Nijmegen and Capgemini.