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 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.

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 reference 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 tradeoff? 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 security 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