Chapter 4

FRAMING ENTERPRISE ARCHITECTURE: A META-FRAMEWORK FOR ANALYZING ARCHITECTURAL EFFORTS IN ORGANIZATIONS

Marijn Janssen

Editors' Preface

As part of EA planning EA programmes have the need to estimate the effort needed. Current literature and practice approaches do not offer much help in this regard. EA effort estimations are at best of experience of the team involved. There is a clearly a lack of scientific / heuristic based approaches on offer. There are several reasons for the current state: (A) Most of the EA discipline development, till date, has focused on frameworks, methodologies, notations, tools, toolkits etc. (B) EA is still evolving, as in, currently the footprint (scope and nature) of the EA programmes are still left to specific needs of the organizations, and (C) Emergence (but current lack) of standards in EA.

The chapter presents an architecture meta-framework that views the architectural elements (subsystems) and the dependencies among the

elements. These linkages captured through a series of layers provide the necessary inputs for coherency.

The meta-framework is comprehensive to the extent that is looks into how it can useful in the 'extended' and 'embedded' mode. It is important to be mindful of the fact that in the embedded mode, EA happens not because of a special programme in the organization, but in the course of regular activities. The EA team / group sets the policies, principles, standards, formats etc. and the organization looks to harvest the usual management artifacts for the purposes of EA.

Even though not mentioned explicitly, the proposed meta-framework is generic enough to be compatible with existing methodologies (like TOGAF ADM) and also provides the capability for further enhancements. Furthermore, the chapter presents scenarios where the framework is expected to perform well and why. This we believe is important to understand the ramifications of adopting the framework for organizations.

Introduction

In the very foundation, Enterprise Architecture (EA) aims to bridge the gap between business and technology and should address the dependencies among a large number of heterogeneous elements (Doucet et al, 2008). Traditionally, the purpose is to effectively align the strategies of enterprises with their business processes and the coordination of their resources (Zachman, 1987). EAs define and inter-relate data, hardware, software, and communication resources, as well as the supporting organization required to maintain the overall physical structure required by the architecture (Richardson et al, 1990). Nowadays, there has been a shift from the foundation architecture, to the extended architecture resulting in the embedded architectures (Doucet et al, 2008).

Enterprise architecture (EA) aims to bridge the gap between business and ICT departments and conceptual and implementation design by defining a systems composition from various viewpoints. Often it incorporates a blueprint of the existing and desired design and an overall plan regarding realizing part of it. Enterprise architecture (EA) lacks a universally accepted definition (Ross, 2003). EA has been characterized as a system of systems (Kaisler et al, 2005), as the “master plan” or “city plan” (Rohloff,

2005) detailing policies and standards for the design of infrastructure technologies, databases, and applications (Bernard, 2004; Ross, 2003). EA efforts are often aimed at creating some kind of coherence and structure in a chaotic environment using systematic approaches (Armour et al, 1999; Doucet et al, 2008). Enterprise architecture models provide ways to deal with the complexity including work (who, where), function (how), information (what) and infrastructure (how to) (Ross, 2003). The main idea of enterprise architecture is that it can be used to guide design decisions and ensures that the dependencies among elements are managed.

The enterprise architecture framework formula specifies how information technology is related to the overall business processes and outcomes of organizations, describing relationships among technical, organizational, and institutional components of the enterprise (Zachman, 1987). While these general frameworks are useful and there exists some literature comparing the frameworks (e.g. Leist et al, 2006; Schekkerman, 2003), very little is known about how they are adapted and used. Furthermore the focus of the efforts is dependent on the way the framework is translated to the organization and the allocated resources. The use and effectiveness is determined by the governance of the architecture. As such, there is a need for a framework that supports the evaluation of the use of enterprise architecture in organizations.

Doucet et al. (2008) argue that coherency management is the primarily outcome of EA efforts. Coherence should ensure that the competitive resources and capabilities of the firm should be "complementary" or "synergistic" and the elements operate in concert. A meta-framework can be used to investigate if the organizational architecture is coherent. In this chapter we will present a framework for analyzing and framing enterprise architecture efforts. The purpose is not to create yet another EA framework. The goal is to help architects and managers to position their EA efforts and to show what elements could be part of the EA efforts. As such the framework remains at a meta-level and can be used to show the focus of the EA and identify elements currently (purposefully) not covered by EA.

Architecture Meta-Framework

Basically, EA is the overview of the enterprise as a whole, from helicopter point of view equipped with x-ray visions in which you not only look at the existing state but also at possible future states. In our view, *enterprise*

architecture is something abstract and remaining at a conceptual level (is an abstract description of reality or an abstract description of wanted realities) and is a frame of reference to guide *design* efforts (design as the creation of an artifact) and at the same time serves as a framework for positioning the design projects. The designs are aimed at improving (parts of the) *infrastructure* (the actual reality, i.e. implementations) and take into account the relationships as depicted by the architecture (or link between strategy and designers). Architecture can be descriptive or prescriptive. Design projects change the infrastructure and therefore the descriptive architecture needs to be updated. The experiences and results of the design projects and resulting infrastructure influence the prescriptive architecture, as new standards, architectural principles, reusable building blocks etc., are created in these design projects.

Any functioning enterprise has either explicit or implicit an 'Architecture'. The progression of EA thought and practice has largely been a process of accumulation, not replacement (Doucet et al, 2008). The foundation architecture is an abstraction of the existing infrastructure and a prescription of the Next Generation of Infrastructures (NGI). Figure 1 schematically depicts this cycle of abstraction, designing and implementation and realization of the NGI, which in turn results to an update of the descriptive architecture and influence the prescriptive architecture. This view closely resembled problem solving cycles in which the architecture is the descriptive and prescriptive conceptual model (Mitroff et al, 1974; Sol, 1982). In many organizations the focus might be on architecture in a descriptive, or a prescriptive sense or on both.

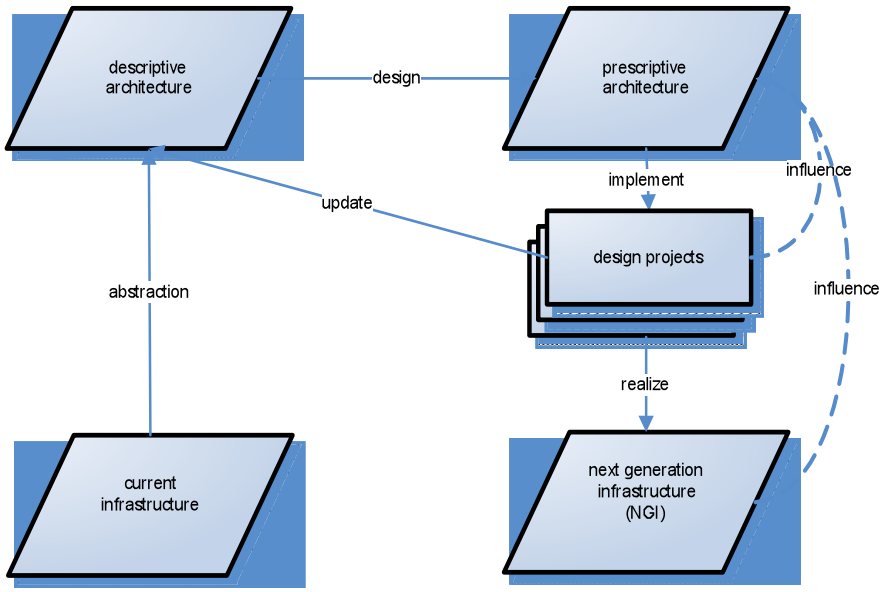

Figure 1: Enterprise Architecture As Abstraction From Reality

Subsystems and the dependencies among subsystems can be viewed to ensure coherency. Malone and Crowston (1994) define *coordination* as the "managing dependencies between activities" and a coordinating mechanism as "the way the interdependent activities and decisions are managed". Generically, architecture is the description of the set of elements and the relationships between them (Armour et al, 1999) and architecture is aimed at creating a coherent and consistent set of relationships among (sub)systems (Doucet et al, 2008; Janssen et al, 2005). The common element is that enterprise architecture refers to a set of interdependent elements described at a certain level of abstraction and the blueprint describes the relationships among the elements. We follow Janssen and Verbraeck (2005) and define *enterprise architecture* as the coordination of subsystems at various levels of abstractions for the purpose of developing the NGI. In this definition, enterprise architecture will be used in both a descriptive or prescriptive manner. Furthermore, coordination can occur at various levels of abstraction and a variety of coordination mechanisms are possible. EA looks at the interrelationship between elements and therefore an essential component of architecture is the description of subsystems and its components.

The subsystems can have a different granularity and be heterogeneous in nature. Often a layered approach is given to organize these subsystems or components in categories of similar objects. For this purpose a layered approach can be used. A layered approach is a systems approach aimed at dealing with the complexity. Ideally, a layered model has the following characteristics (e.g. Stallings, 2006):

1. Each layer performs a cohesive or closely related set of functions
2. Higher layers use services provided by the lower layers
3. Layers are sufficiently loosely coupled to allow changes in one layer without affecting other layers

The layered approach is nowadays motivated by Service-Oriented Architectures (SOA). The basic idea of SOAs is to decompose a system into parts that are made accessible by services, to design these services individually and to construct new systems using these single services (Cherbakov et al, 2005). Each component provides a (set of) functions accessible by one or more services for use by other components. Service-orientation makes the architecture agile, as companies can easily substitute components without having to change the interface or other components. Adopting service-orientation offers many benefits to enterprises, making it possible to create services that are modular, accessible, well-described, implementation-independent and interoperable (Fremantle et al, 2002, p. 80). A variety of services can be covered including business services and low-grained software services. The type of services can be divided in layers. In EA, layers can be used to group and structure similar kind of items, including services. Each layer is dependent on other layers and the dependencies within a layer and between layers should be addressed. Furthermore, standards, models and architectural principles can be positioned in each layer. In this way, the EA design process becomes a process of defining layers, as well as the services provided in each of the layers.

The premise of SOA is that it has tremendous potential and can offer improved efficiency, reduce development costs and risks, create agility and flexibility and to reuse existing systems (Khoshafian, 2006; Krafzig et al, 2004; McGovern et al, 2006). Services can occur at various levels of granularity and can be composed of other services. For example, a business service might be the handing of claims which is created by a business process consisting of an application service for identifying the user, and an information service for obtaining customer information. The latter uses in turn an infrastructure service for secure communication. In

this way services can occur at various layers including business, business process, information, application and infrastructure and become the main focus of the EA efforts.

Our layered model is especially focused on characteristics one of Stallings (2006), layers should describe a set of cohesive or closely related set of elements in order to create structure in a chaotic environment. Characteristics two and three of Stallings can be used as architectural principles guiding the development of a prescriptive architecture, as the current infrastructure might not adhere to these two characteristics. In this way a complex system can be decomposed in elements categorized using some criterion.

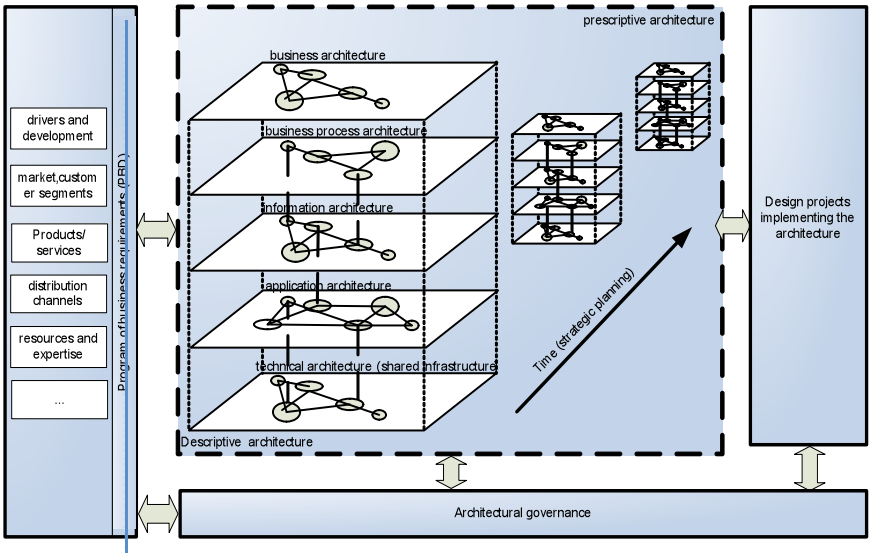

Figure 2: Overview Architectural Meta-Framework

Tapscott and Caston (1993) identified a number of categories of business drivers in which the ICT and business contexts need to be balanced. These categories are adopted as layers in our framework and include the organizational, business process, information, application, and technical architecture layers. EA efforts can be organized using the five layers, depicted in the middle of Figure 2, that constitute the core elements of our meta-framework.

Each layer and even each system can be further decomposed. Functional decomposition is a common approach found in systems theory (Sage et al, 2000). A system can be decomposed into parts, developed and only

thereafter integrated with the other systems based on the architecture. Each function has inputs and outputs and the hierarchical decomposition describes the transformation from available inputs to outputs. Layers might be further decomposed in sub layers. For example the information architecture layer might contain the sub layers product information, management information and operational process information. Furthermore, elements in the layers might be further decomposed into sub-elements. In this way a more detailed architecture with various levels of abstraction can be created.

The layers provide a coordination view and are aimed at creating coherency however, they do not include various viewpoints on the system (Architecture Working Group, 2000; Zachman, 1987). A view is "a representation of a whole system from the perspective of a related set of concerns" (Architecture Working Group, 2000) and a viewpoint "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 (Architecture Working Group, 2000). Examples of a view are the user, developer or security, governance and adaptability view and a viewpoint establishes the rules conventions by which a view is created, visualized and analyzed. These views typically go beyond a single layer and can be added as a vertical column which goes through the layers. We did not add views to the figure in order to keep it simple and understandable.

EA is often aimed at creating a common operational picture and a shared vision of the enterprise and its environment. An architecture can be descriptive, prescriptive or both. In a descriptive sense, an architecture shows the existing relationships among the elements which can be used to analyze the weaknesses and opportunities. In a prescriptive sense it can be used as a concept of the desired future situation and operationalized as a blueprint that needs to be realized within a certain timeframe. If both descriptive and prescriptive architectures are available, a growth path or strategic roadmap from the current to the envisioned future situation can be realized, as depicted by the time (strategic planning) axis of Figure 2. The need of realizing a new situation results in implementation projects, which are shown on the right hand of the figure, which in turn influence the descriptive and prescriptive architecture as already explained using Figure 2.

Figure 2 shows that the enterprise architecture in the middle is closely related to the environment, its governance and to implementation projects.

Architectural governance mechanisms are necessary to make the architecture work. It should ensure that the architecture is actually used. The environment, governance and implementation elements can change the enterprise architecture descriptions and in turn, the enterprise architecture can influence these three elements. As such, these other three elements are considered to be part of the meta-framework.

Environment and Strategy

The b*usiness and multi-actor environment* contain the situational factors influencing the architectures. These include elements like the products, market segments, market and technology developments, but also the resources and capabilities that can be used for realizing the architecture and resulting infrastructure. This element contains all information necessary as input for engineering the EA.

The outcome of this element is a ***program of business demands*** **(PBD)** that is used as a start for defining the desired enterprise architecture (Nijhuis, 2006). The PBD serves as a bridge between the Business environment and strategic objectives on the one hand and the enterprise architecture on the other hand. The PBD has to provide a concrete indication to what requirements the EA has to meet. Thus, the PBD is a guideline for the development of the architectures, and it will be used in retrospect to test their quality. Further, the requirements should be consistent and coherent and as such contains trade-offs, e.g. preference of low costs over high service levels or proven technology over innovativeness. The PBD focuses completely on the desired situation.

Often the PBD consist of a collection of architectural principles. *Principles* are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission (Perks et al, 2002; Richardson et al, 1990). Principles can be described 1) Name, 2) Statement, 3) Rationale and 4) Implications (Perks et al, 2002). These principles provide the basis for decision-making throughout all departments, organizations and projects.

Architecture Layers

The PBD needs to be operationalized in a conceptual architecture. Architecture layers are the core of the meta-framework as it can be used to analyze the relationship within and between architecture layers and can be

used to position systems, standards, architectural guidelines and other architectural methods and tools. Architecture layers are aimed at relating the various systems within and between layers. The highest layers are focused on the business, whereas, the lower layers are more focused on the technical aspects. As such the layers can be used to align the business and IT with each other by analyzing the coherence between business and IT using the layers. The layers can be designed using a top-down approach starting from the strategy or PBD, a bottom-up approach starting with the existing systems, a middle-out approach consisting of a combination of bottom-up and top-down approach or by first creating the vision on the architecture and creating a prescriptive architecture.

Business Architecture

The organizational or business architecture layer is aimed at describing the arrangements of the responsibilities around the most important value-creating activities. This layer describes the decomposition of the enterprise in different responsibilities and the coherence among those responsibilities (Versteeg et al, 2005). This layer deals with who is responsible for a specified organizational part and decouples the business domains from each other. Organizational parts typical provide service to one or more other parts. As such, each organizational part should have clear interfaces and service level agreements with the other parts to ensure coherence and smooth functioning of the complete organization.

The organizational level is related to the business process level, as it should be clear who is responsible for the continuous improvement of the end-to-end process. After all, many business processes will go beyond traditional departmental boundaries. This level is related to the information layer as accountability for information and information quality should be part of allocation of responsibilities. Furthermore, in this layer it should be described who is responsible for controlling and maintaining applications in the application layers and ensuring service levels are met.

Business Process Architecture

The business process architecture layer is a collection of business processes and the relationships among them. It describes the functional composition of the business into process flows. Business process architecture can serve to define the scope of design projects by showing the boundaries of the domain of interest and the needed output of a process.

A business process is a collection of interrelated tasks which solve a particular issue. There are at least three types of business processes:

1. Management and control processes: governing the operation of a system;
2. Operational processes: constitute the core business and are aimed at adding customer value;
3. Supporting processes: support the core processes.

A set of business processes begins with a customers need and ends with that need being fulfilled. A business process can be further decomposed into sub-processes. Sub processes can be triggered by external or internal events. Usually these processes are the result of customer interactions, but business process itself can trigger another one or might periodically start (for example invoicing).

The analysis of business process architecture typically includes the mapping of the main processes triggered by external events. A business process can be further specified into tasks in design and implementation projects. Many reengineering projects start with the business processes and this phenomenon is called Business Process Reengineering (BPR) (Hammer 1990; O'Neill et al, 1999). Each business process can be related to the resources it consumes and the products it produces. Resources include human capacity and information. The products produced by an activity can themselves be consumed as resources by other activities.

Information Architecture

The information architecture layer is aimed at describing the information assets aimed at storing, processing, reusing and distribution of information across information resources to fulfill the stakeholder needs. Information architecture is the organization of information to aid information sharing among actors. Information architecture is the pivot between the business processes and applications, business process use and process information and information is stored in applications.

Information stewardship is often used as a principle for organizing this layer. Information stewardship is making departments or person accountable for information and its quality. The steward oversees information throughout its life cycle. If the information steward also owns the information a vital records registry is created. In such registries

information is stored and maintained and all other departments have to make use of this register and are not allowed to store this information in their own systems.

Application Architecture

The application architecture layer contains the conceptualization of the software applications, components and objects, and the relationship between these parts. As such, this layer is typically further decomposed in different types of applications (business process management, document management, office applications etc). In addition this layer should guide application selection and the integration of applications with each other.

Architecture-based application planning should take place at the portfolio level (Hamilton, 2004). "Information technology portfolio management is the management of IT as a portfolio of assets similar to a financial portfolio aimed at to improving the performance of the portfolio by balancing risks and returns" (Jeffery et al, 2004). Nowadays it is sometimes argued that the attention shifts from application to service portfolios, as it can fasten implementation and lower costs, as it enables the rapid composition of service provisioning processes from reusable components (Janssen et al, 2006). As such the management and development of services become the key focus.

Technical Architecture (or Shared Infrastructure)

The Technical architecture or shared infrastructure is about generic facilities, used by many other systems. It is about functionality that is a common need of many different systems and should not be mixed up with the terms "current or next-generation infrastructure". The shared infrastructure contains the network infrastructure, the operating systems and other generic services and facilities providing functionality that is used by many systems. This layer is the foundation for the creation of the application architecture. Over time more and more systems become part of this layer, as many current efforts are targeted at developing reusable building blocks and to assemble new systems using these building blocks.

Implementation, Control and Maintenance

The architecture needs to be implemented by projects and projects change the architecture. In this way, these two parts are mutual dependent on each other. Even sub products, like the realization of a component, might alter the architecture, as a component can be used as a building block in the architecture. Architecture is never the end. Architectures need to be implemented, controlled and maintained in an efficient and effective way. Only in this way "EA can be considered the ongoing, overarching method for abstracting, analyzing, designing, and re-engineering new and existing enterprises" (Doucet et al, 2008). The projects contribute to the realization of the EA and need to take into account the architecture. For this purpose, architectural governance is needed.

Architectural Governance

Letting an architecture work, depends on people, the creating of commitment and mutual understanding and trust. People should have incentives to adhere to the principles and rules and be motivated to make use of the architecture. Architectural governance is a form of IT governance that can be described as "the structure of processes to direct and control the enterprise in order to achieve the enterprise's goals by adding value while balancing risk versus return over IT and its processes" (Peterson 2004). Governance represents the framework for decision rights and accountabilities to encourage desirable behavior in the use of resources (Weill et al, 2002). Enterprises generally design three kinds of governance mechanisms: (1) decision-making structures, (2) alignment processes and (3) formal communications (Weill et al, 2005). These mechanisms should ensure that the architecture is known and disseminated among the organization and is necessary for embedding architecture in all aspects of the organization, in this way creating an embedded architecture (Doucet et al, 2008).

IT often requires major investments for organizations and can comprise hundreds or event thousands of projects running simultaneously across departments. Therefore applications or service portfolios can be an important government instrument. Furthermore service portfolios form the bases for planning further development. In this way a portfolio can help to develop a growth plan determining which services should be developed by

whom, when new releases of existing services should be developed, the expected functionality, performance and characteristics of services, who should maintain the services or how the services should be sourced.

Evaluating an Enterprise Architecture

The meta-framework was used to evaluate the architectural efforts of a large organization. This organization consists of several departments, one department responsible for all front-office activities, several back-office departments and several supporting departments. In the front office most of the applications were developed in-house, whereas the back-office application were bought on the market and provided by proprietary software vendors. The type of research undertaken was action research, as the researchers became involved in the application of the framework. The company wanted to assess its architectural effort and know how it should be expanded to make architecture an integral part of the organization.

Relation to the environment and strategy

It was found that there were clear strategic objectives; however there was no such thing as a PBD that was used to translate the strategy into a set of requirements on the prescriptive architecture. This resulted in limited coherence among decisions. For example, it was not clear if open source software was favored over vendor developed software and as a result both types of software could be found within the company. Another example is that it was unclear if the use of open standard was more important than the cost of buying a software package. As a result these types of choices were made during the execution of projects. Furthermore, the priority might be different for the front and back office, as the front office applications were primarily developed in-house and the back office applications on the market.

Another example is that the ambition of the board was to integrate with all the systems of all their trading partners. There was no prioritization based on the easiness to integrate systems or transaction volumes and it was not clear if it would be feasible to integrate with all systems. It could be easily calculated that integration with all systems would be too expensive and never be profitable. It was concluded that the PBD was necessary to get more grip on the architecture. In the PBD the front and back office domains should be addressed separately and clear objectives should be stated and prioritized.

Focus on layers

In this company EA was primarily used to show the dependencies among applications and provide a means for understanding and management of the complex application landscape. Standards for ensuring interoperability were developed, identical and similar systems were identified and rationalized, and reusable software components that could be accessed as web services were identified. The scope of the architectures needs to be balanced, as not everything can be done when having limited resources and capabilities, therefore the focus on the application layer was viewed as the right decision by the management. Although the focus on the application layer in the architectural framework is a logical one and can result in large cost savings, the comparison of the relationships with the other architecture layers showed that the responsibilities for developing reusable applications were not allocated. This was left to the projects, which were primarily aimed at developing systems for their own purpose and were not focused on designing reusable systems. The relationship among the application and organizational layer was ignored and a recommendation was to better allocate resources.

Architectural principles were primarily aimed at supporting system design and included principles like the reuse of existing information, defining all data elements and storing it, always allocations of ownerships of processes and data to a certain owner, and the owner is responsibility for ensuring the quality. Furthermore, the principles were only stated and the rationale and implications of the principles were not described, as suggested by Perks and Beveridge (2002).

Implementation, control and maintenance

Like in most other organizations, the design of an EA was a balancing act as time as resources were limited and circumstances changed continuously. The level of abstraction of the EA needed to be balanced. This included the granularity of the descriptions and prescription of applications. Applications can be spelled out or the EA can only give guidelines at a high level of abstraction. The architects had the tendency to spell out the system in much detail. This did have the advantage that the right kind of granularity was taken for developing reusable components. By having this level of detail the architecture became less agile and vulnerable to changes, as it provided less freedom to the system developers, required more resources from the architectural department, needed a long time to develop

and needed to be changed if new applications would be needed. Instead of making clear decisions concerning the preferred direction and setting the constraints, the architects took over the roles of the designers.

Architectural governance

There were several departments that have their own way of working and defined their own standards; as such the architectural governance was weak. There are no clear decision-making structures and alignment processes. There was formal (and informal) communications to ensure that the architecture was known by the staff and that the staff would understand the role of the architecture. The interviews showed that architecture was primarily viewed as a playing tool for the architectural department, which could be easily ignored. This resulted in a continuation of the interoperability problems and economies of scales could not be accomplished as facilities were not shared and components were not reused.
As a portfolio instrument a list of applications was created. Every two months an application portfolio team met and decided about the buying of new applications. The primarily focus of the team was on standard software packages like drawing software. Reusable services were not integrated in this portfolio and the portfolio was not used as part of the software development process.

By using the framework it was illustrated that some elements that are typically part of architectural efforts were not addressed in the organization. The framework is not normative in the sense that this would automatically imply that the missing elements should be addressed. Addressing all elements might not be possible given the limited amount resources or simply not necessary. EA is often focused on solving certain problems and for solving these problems certain elements can be addressed. In this company, the management did not mind that the focus of the EA was on the application layer. They viewed EA as being an instrument for getting grip on the applications, and decided to reconsider this once the other elements were addressed. The managers did mind that there was no PBD, the lack of governance, especially the bypassing of the EA in projects. The result of this analysis was that elements were included in the year thereafter.

Conclusion

In this chapter a meta-architecture framework was presented aimed at helping architects and managers to position their EA efforts and to show what elements could be part of the EA efforts. The framework remains at a meta-level and can be used to show the focus of the EA and identify elements currently, which could be purposefully, not covered by EA. This was illustrated using a case study.

The core elements of our meta-framework consist of a layered model that can be further decomposed in sub-layers and systems. The layers can group similar elements and the layered models can be used to show the coherence among elements. Components accessible as services can be positioned in the layered model. By substituting components without affecting the service interface the EA becomes agile. The meta-framework shows that both descriptive and prescriptive architectures can be available and growth path or a strategic roadmap from the current to the envisioned future situation can be determined as part of the architectural efforts. On each layer standards, models and architectural principles can be defined and positioned. Another element includes the multi-actor environment and organizational strategy resulting in a program of business demands guiding the decision-making. PBD is crucial is this is the link with the organization strategy and determines the priorities of goals for the architecture. The last elements are the design projects implementing the architecture and architectural governance. Projects update the architecture and governance is necessary to ensure that the architecture will be used and is complied with. Architectural governance is necessary for creating an architecture embedded in the business.

The action research showed that the meta-framework can be used to evaluate and position architectural efforts within organizations. For organizations, the design of an EA is a balancing act as time and resources are limited and circumstances change rapidly. At least, the level of detail and the focus of the architectural efforts need to be chosen. The case study shows that the meta-framework presented in this chapter is a suitable instrument for evaluating the architectural efforts of an organization and to determine if an organization is doing the right things.

References

Architecture Working Group. (2000). IEEE Std 1471-2000 Recommended Practice Architectural Description of Software-Intensive Systems. IEEE Software Engineering Standards Committee.

Armour, F.J., Kaisler, S.H., and Liu, S.Y. (1999). A big-picture look at Enterprise Architecture. IEEE IT Professional. Volume 1, Number 1. pp. 35-42.

Bernard, S.B. (2004). An introduction to Enterprise Architecture. AuthorHouse: Bloomington, IL.

Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., and Rackham, G. (2005). Impact of service orientation at the business level. IBM Systems Journal. Volume 44, Number 4. pp. 653-668.

Doucet, G., Gøtze, J., Saha, P., and Bernard, S. (2008). Coherency Management: Using Enterprise Architecture for Alignment, Agility, and Assurance. Journal of Enterprise Architecture. Volume 4, Number 2. pp. 9-20.

Fremantle, P., Weerawarana, S., and Khalaf, R. (2002). Enterprise services. Examine the emerging files of web services and how it is integrated into existing enterprise infrastructures. Communications of the ACM. Volume 45, Number 20. pp. 77-82.

Hamilton, D. (1999). Linking strategic information systems concepts to practice: System integration at the portfolio level. Journal of Information Technology. Volume 14, Number 1. pp. 69-82.

Hammer, M. (1990). Reengineering work: Don't automate, obliterate. Harvard Business Review. Volume 68, Number 4. pp. 104-112.

Janssen, M., and Feenstra, R. (2006). From application to Web service portfolio management: Concepts and practice. European Conference on e-Government (ECEG). Philipps University, Marburg, Germany. pp. 225-234.

Janssen, M., and Verbraeck, A. (2005). Evaluating the Information Architecture of an Electronic Intermediary. Journal of Organizational Computing and Electronic Commerce. Volume 15, Number 1. pp. 35-60.

Jeffery, M., and Leliveld, I. (2004). Best Practices in IT Portfolio Management. MIT Sloan Management Review. Volume 45, Number 3. pp. 41-49.

Kaisler, S.H., Armour, F., and Valivullah, M. (2005). Enterprise Architecting: Critical problems. Proceedings of the 38th Hawaii International Conference on System sciences, Hawaii, USA.

Khoshafian, S. (2006). Service Oriented Enterprises. Auerbach Publication: New York, NY.

Krafzig, D., Banke, K., and Slama, D. (2004). Enterprise SOA. Service Oriented Architecture Best Practices. Prentice Hall PTR: Indianapolis, Indiana.

Leist, S., and Zellner, G. (2006). Evaluation of Current Architecture Frameworks. Proceedings of the 2006 ACM symposium on Applied computing. ACM, Dijon, France.

Malone, T.W., and Crowston, K. (1994). The interdisciplinary study of coordination. ACM Computing Surveys. Volume 26, Number 2. pp. 87-119.

McGovern, J., Sims, O., Jain, A., and Little, M. Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations. Springer, 2006.

Mitroff, I.I., Betz, F., L.R.Pondy, and Sagasti, F. (1974). On Managing Science in the System Age: Two schemes for the study of science as a whole systems phenomenon. TIMS Interfaces. Volume 4, Number 3. pp. 46-58.

Nijhuis, R. (2006). Enterprise Architectures as a Guideline for the Implementation of Strategy. Realising Sustainable Performance Improvement for Pension Providers and Insurance Companies DCE Holding b.v., Schiphol.

O'Neill, P., and Sohal, A.S. (1999). Business Process Reengineering A review of recent literature. Technovation. Volume 19, Number 9. pp. 571-581

Perks, C., and Beveridge, T. (2002). Guide to Enterprise IT Architecture. Springer: New York, NY.

Peterson, R. (2004). Crafting Information Technology Governance. Information Systems Management. Volume 21, Number 4. pp. 7-22.

Richardson, L., Jackson, B.M., and Dickson, G. (1990). A Principle-Based Enterprise Architecture: Lessons from Texaco and Star Enterprise. MIS Quarterly. Volume 14, Number 4. pp. 385-403.

Rohloff, M. (2005). Enterprise architecture: Framework and methodology for the design of architecture in the large. European Conference on Information Systems.

Ross, J. (2003). Creating a strategic IT architecture competency: Learning in stages. MISQ Executive. Volume 2, Number 1. pp. 31-43.

Sage, A.P., and Armstrong, J.E. (2000). Introduction to Systems Engineering. John Wiley & Sons: New York, NY.

Schekkerman, J. (2003). How to Survive in the Jungle of Enterprise Architecture Framework: Creating or Choosing an Enterprise Architecture Framework Trafford.

Sol, H.G. (1982). Simulation in Information Systems Development. University of Groningen, Groningen.

Stallings, W. (2006). Data and Computer Communications. Prentice Hall

PTR: Indianapolis, Indiana.

Tapscott, D., and Caston, A. (1993). Paradigm Shift - The New Promise of Information Technology. McGraw-Hill: New York, NY.

Versteeg, G., and Bouwman, H. (2005). Business architecture: A new paradigm to relate business strategy to ICT. Information Systems Frontiers. Volume 8, Number 2. pp. 91-102.

Weill, P., and Ross, J. (2005). A matrixed approach to designing IT governance. MIT Sloan Management Review. Volume 46, Number 2. pp. 26-34.

Weill, P., and Vitale, M. (2002). What IT Infrastructure Capabilities are Needed to Implement E-Business Models. MISQ Executive. Volume 1. Number 1. pp. 17-34.

Zachman, J.A. (1987). A Framework for Information Systems Architecture. IBM Systems Journal. Volume 26, Number 3. pp. 276-292.