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.

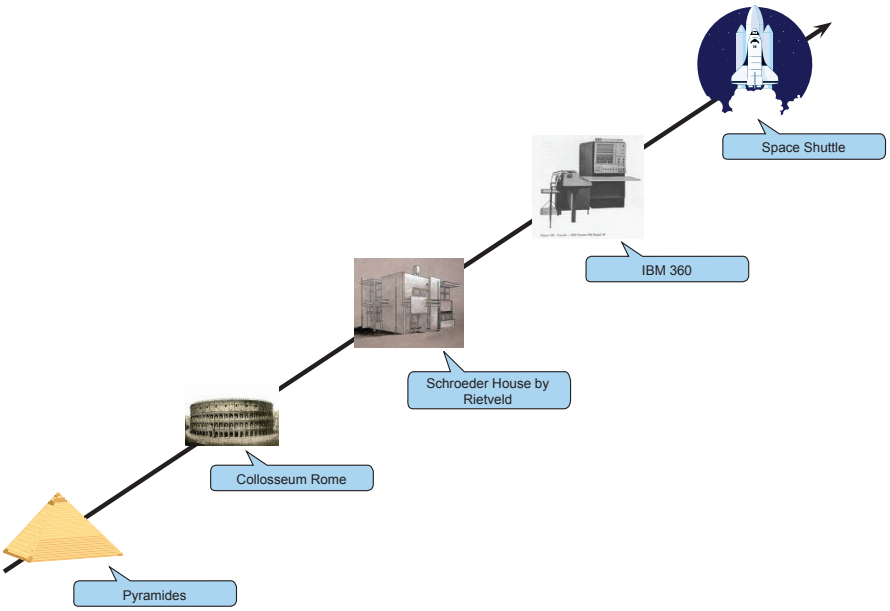

Fig. 3.1 Evolution of constructions

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

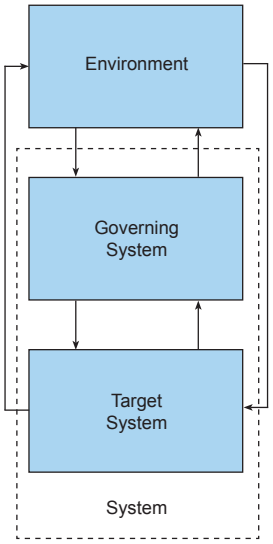

Fig. 3.2 The basic governance paradigm

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*.

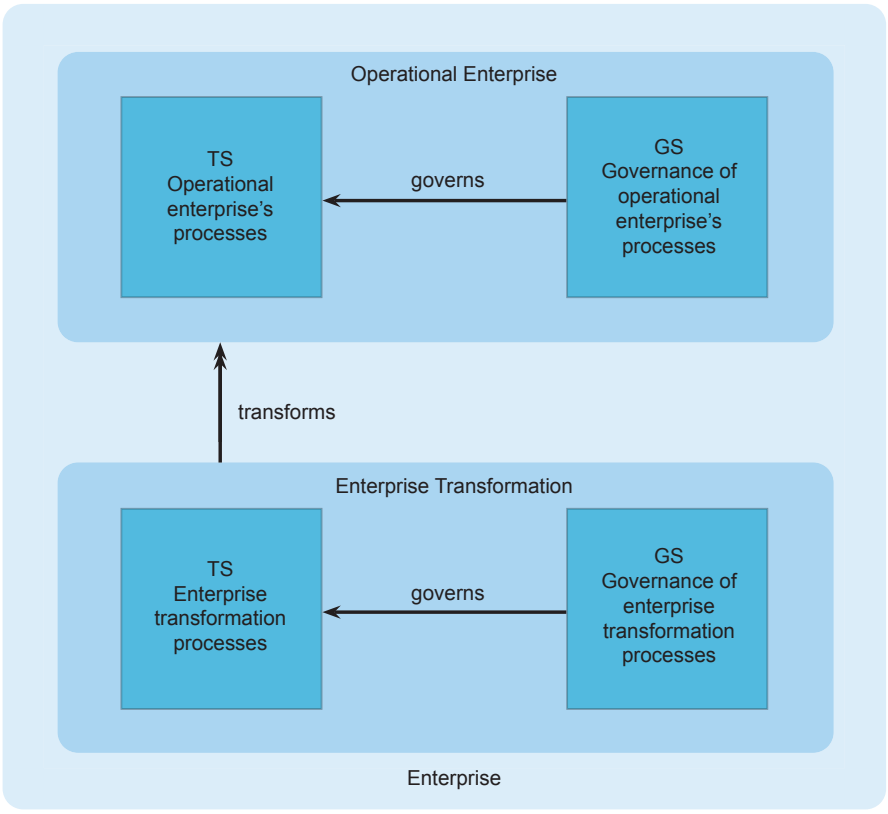

Fig. 3.3 Governance of an enterprise's transformation

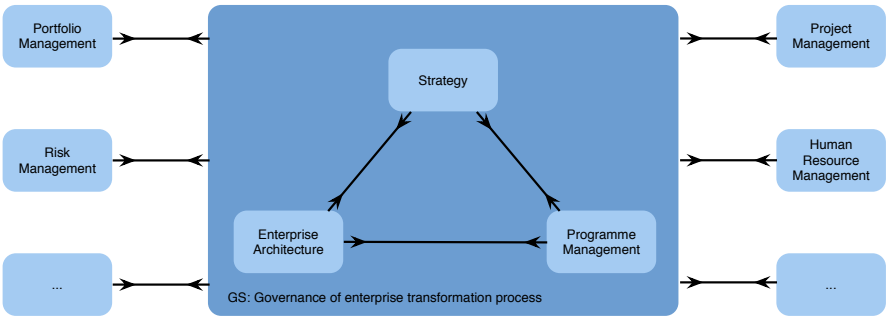

Fig. 3.4 The role of enterprise architecture

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.

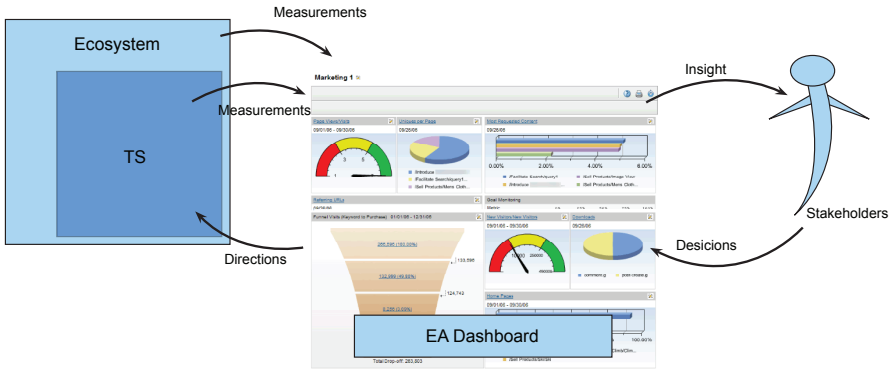

Fig. 3.5 Enterprise architecture on a dashboard

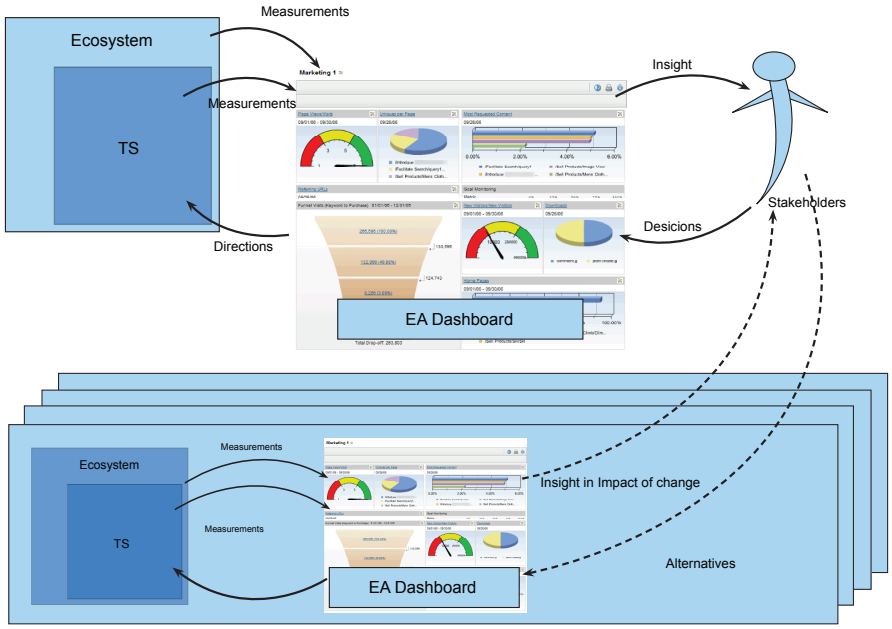

Fig. 3.6 Pro-active governing of transformation process

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.

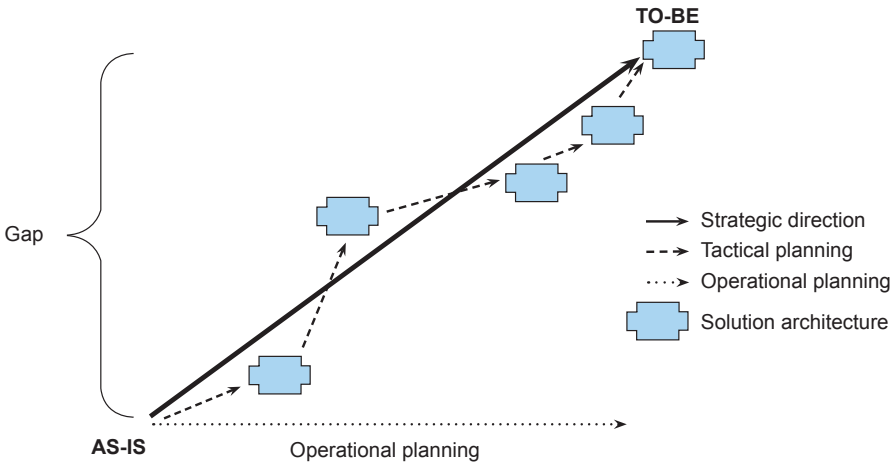

Fig. 3.7 Applications for enterprise architecture

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*);

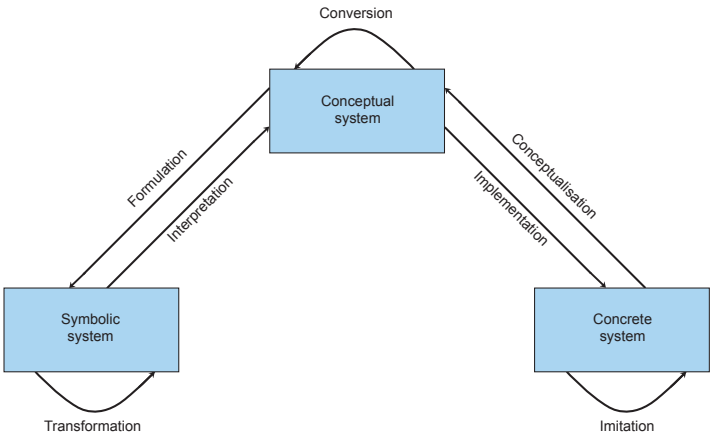

Fig. 3.8 Three types of systems

- 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.