25

Enterprise Architecture

25.1 Introduction .................................................................. 25-1

25.2 Positioning Enterprise Architecture ............................. 25-2

EA Processes and Services • Role of IT Governance • Business Architecture • Application Architecture • Data/Information Architecture • Technology Architecture

25.3 Reasons for Enterprise Architecture .............................. 25-5

EA Drivers • EA Benefits and Value Contributions

25.4. EA Frameworks .......................................................... 25-6

Zachman EA Framework • Generalized Enterprise Reference Architecture and Methodology • ARIS • TOGAF • Other Frameworks

25.5 EA Modeling .............................................................. 25-12

25.6 Practical Aspects of EA .............................................. 25-12

EA Maturity • EA Tools

25.7 Critical Aspects ........................................................... 25-13

25.8 Summary ................................................................. 25-13

References ...................................................................... 25-14

Martin Meyer
Dublin City University

Markus Helfert
Dublin City University

25.1 Introduction

Since the inceptions of information technology (IT) and information systems (IS), companies have experienced a constant increase in size, scope, and complexity of enterprise IS. To manage and organize these systems, logical constructions and representations in the form of models were needed, and an architecture approach to IS was developed in response to these challenges. Architectures are used commonly in IT and IS domains to construct blueprints of an enterprise for organizing system components, interfaces, processes, and business capabilities, among others. Architectures are discussed extensively, describing various aspects of IS. Architectures in the wider context of IS are often used to model aspects of a system, especially computer, network, software, application, service-oriented, business, and project-development architectures, among others. Thus, the term architecture plays a central role in IS and IT and can be defined generally as “the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution” (IEEE Computer Society, 2007).

Combined with increased importance of IS in the 1990s (Morton 1991), information system architectures—and later enterprise architectures (EA)—became important. Yet, this is still a broad and multifaceted domain with various viewpoints. Over the years, many EA frameworks, modeling concepts, and tools were proposed, including the Zachman framework (Zachman 1987), Department of Defense Architecture Framework (Chief Information Officer U.S. Department of Defense 2010), Federal Enterprise Architecture Framework (Office of Management and Budget 2012), The Open Group Architecture Framework (TOGAF) (The Open Group 2011), and Architecture of Integrated
Information Systems (ARIS) (Scheer 1992). The purpose of this chapter is to describe EA from a business and IS viewpoint as well as review prominent EA frameworks. A common concept found among all frameworks is the conceptualization of strategic business decisions in systems design choices, in which EA aims to support and enable transformations within an organization while aligning business and IT. To reduce complexity, many frameworks use views to describe elements of architectural content (e.g., strategy, information, process, service, and technology). Each view is meaningful to various stakeholder groups. The intent of EA is to determine—from various stakeholder viewpoints—how an organization can most effectively achieve current and future objectives in the form of transformations. Strategic options are often translated into systems design options, and EA helps manage this process carefully.

The terms enterprise architecture (EA) and enterprise architecture management (EAM) are often used interchangeably (Ahlemann et al. 2012b), though the scopes of the two are different. EA is concerned with creation in the form of design and development of architectural blueprints and artifacts; EAM is concerned with the sustained management of such artifacts using lifecycles. Consequently, EAM is a process to plan, control, govern, and evolve an enterprise’s organizational structure, business processes/services, and technical infrastructure in accord with a vision and strategy. We view EAM as part of EA, and hence, refer only to the EA discipline as whole. EA is concerned with business process integration and standardization (Ross et al. 2006), and thus, business processes are at the core of EA. We view EA from a business process and modeling perspective, though EA often includes a large variety of IS-related viewpoints and approaches in practice.

### 25.2 Positioning Enterprise Architecture

EA evolved into its own research domain, and various ways of positioning and organizing EA emerged (Ahlemann et al. 2012b; Dern 2009; Hanschke 2010; Op ’t Land et al. 2009; Ross et al. 2006; Scheer et al. 2000; Schmidt et al. 2011; The Open Group 2011). Early work on architectures model aspects of systems, often referred to as IS or system architectures. These architectures do not emphasize strategic directions or business aspects strongly (Kim et al. 1994). Given a need for IT and business alignment and coming from early system-oriented concepts, a widely adopted approach emerged that organizes EA along various organizational layers and emphasizes an IT business alignment paradigm. The purpose is to relate strategic aspects to application and technology. Layering helps researchers and managers understand and describe the scope and function of EA as they relate to boundary points from business strategy to technical infrastructure. Figure 25.1 illustrates both architectural layers and the contexts with which they interact. Usually, strategic planning serves as input for EA, especially when EA is viewed as a strategy-driven enterprise function. Core layers represent business architectures (BAs), application architectures (AAs) (including data/information architectures), and technology architecture, all used with EA processes and services to design, develop, govern, and manage them. EA produces some form of content or output such as artifacts (Winter et al. 2006) and principles (Greefhorst et al. 2011). In addition, EA uses frameworks and tools.

We view EA from a service perspective; each layer offers a service to the business, and the service-oriented paradigm occurs in each architectural layer. For example, the technology architecture offers infrastructure services in the form of hardware and networks. The application architecture provides services centered on software applications and data. The BA is concerned with business processes and services. The BA, through business processes, emphasizes the dynamic aspects of workflows and activities supported by application components and infrastructures. Examples in the literature present simple, three-layered frameworks with which to view EA (Hasselbring 2000; The Open Group 2012) to multilayered EA frameworks (Winter et al. 2006). Approaches that include enterprise strategy as a separate layer are also documented (Godinez et al. 2010). Whether or not strategy is a dedicated EA layer,
it initiates a discussion of whether EA is solely an IT or also a business function. Due to the strategic importance of business–IT alignment (BITA), we argue EA is a hybrid enterprise function (Ahlemann et al. 2012b). In the following sections, we describe EA processes and services, the role of IT governance, and core architectural layers.

25.2.1 EA Processes and Services

EA services provide benefits to stakeholders within an organization and may consist of several EA processes. EA service model (Robertson 2008) can be introduced to document the service delivery of EA. An example is described in Table 25.1. Notably, there are more processes in relation to these fundamental EA services, and EA services need to be adapted to specific enterprise contexts.
25.2.2 Role of IT Governance

In the context of EA, the role of IT governance is important since it provides a means for management to direct and govern IT toward desired outcomes (IT Governance Institute). IT governance is a tool that provides necessary means to achieve strategies and goals. EA is crucial for effective IT governance because it provides methods to measure success (Weill et al. 2004). However, a holistic model that describes IT governance in an enterprise context has not surfaced due to the complexities and diversity of perspectives inherent in the domain (Rüter et al. 2006). EA governance is a subdiscipline of IT governance that focuses on the principles, decision rights, rules, and methods that drive architecture development and alignment in an organization (Greefhorst et al. 2011). EA governance is concerned with standardization and compliance with current and future practices.

25.2.3 Business Architecture

BA, the top layer of EA, typically contains business processes/services, organizational structures (including roles and responsibilities), and value drivers, which are aligned to a strategy divided into goals and objectives (The Open Group 2011; Versteeg et al. 2006). BA marks the foundation of effective BITA, and sample scenarios in a BA context include (The Open Group 2010) the following:

- Mergers and acquisitions planning and developments
- Business unit consolidations
- New product and service rollouts
- Introduction of new lines of business
- Consolidating suppliers across supply chains
- Outsourcing business functions
- Divesting business lines
- Management changes
- Regulatory compliances
- Operation cost reductions
- Federated architecture alignments in government
- Business transformations
- Entering international markets

<table>
<thead>
<tr>
<th>TABLE 25.1 Example EA Service Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>EA Services</td>
</tr>
<tr>
<td>EA processes</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Roles</td>
</tr>
<tr>
<td>Outcomes (deliverables)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Benefits</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
25.2.4 Application Architecture

AAs deal with combined logical capabilities, developing specifications for individual applications including interactions and relationships to business processes. Applications offer services to other business units. Within AA, service-oriented architectures (SOA) (Erl 2007; Krafzig et al. 2007) and cloud computing (Mahmood 2011) are discussed often. These emerged as recent EA paradigms, besides classical enterprise application integration challenges.

25.2.5 Data/Information Architecture

We argue data and information architectures (DA/IA) are situated as part of the AA layer, due to its importance and relation to applications. This does not mean this sub-layer is unimportant or can be seen in isolation; in fact, many view it as the core of an enterprise relating to multiple layers of EA, referred to as the data-centric view. Information consists of business objects such as orders and production plans, and includes raw information gathered throughout the company by means of performance measures (Neely 2004) and translating it into meaningful information as a key objective of EA (Godinez et al. 2010). In this way, DA/IA relates to business processes and translates information into a business context. Consequently, it is an important EA aspect because it has significant impacts on the business, and quality of data and information is particularly important (Borek et al. 2012). It comprises the logical, physical, and management views that provide sufficient information to form a basis for communication, analysis, and decision-making (Allen et al. 1991; Xie et al. 2011). For example, EA is concerned with data models, logical and physical models, data flows and transformations, metadata, and standards on how information should be stored and exchanged.

25.2.6 Technology Architecture

The technical layer of EA is concerned with an underlying infrastructure (i.e., hardware and system-level software). Hardware includes servers and network infrastructures, and all of their components. Applications are built using appropriate infrastructures, and hence, this layer provides required infrastructure services. An example service at this layer is the provisioning of a server. Technology architecture offers infrastructure services to applications; notably a database is a type of infrastructure.

25.3 Reasons for Enterprise Architecture

To provide a review of reasons to employ EA, we examine various EA drivers and key EA applications, and we describe benefits and common outputs of EA.

25.3.1 EA Drivers

EA drivers are the primary reasons for enterprises to employ EA, and one of the most important drivers is BITA (Schönherr 2008). Critical aspects are described in detail when aligning business and IT (Henderson et al. 1993; Luftman 2003; Mahr 2010; Pereira and Sousa 2005). Cost reductions and managing complexities are other important EA drivers. For example, dismantling legacy systems and components reduces costs, and EA reduces complexity by providing a streamlined and manageable system landscape described through various models. In addition to these important and frequently mentioned drivers, some external drivers do not originate within the enterprise. An example is regulatory compliance requirements issued by legal organizations or governments to which the enterprise must adhere.
25.3.2 Benefits and Value Contributions

Outcomes and benefits depend on context and the extent to which an enterprise uses EA. Usual EA output is described in the form of principles, models, and architecture views. Less tangible outputs in the form of common understanding and improved communications are benefits of EA. In the literature, a number of researchers discuss EA contributions in terms of direct benefits and business value (Ahlemann et al. 2012b; Meyer et al. 2012; Ross et al. 2006; van Steenbergen et al. 2011). We summarize some of the typical EA benefits in Table 25.2 (Ross et al. 2005).

<table>
<thead>
<tr>
<th>TABLE 25.2 Benefits of EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology-related benefits</td>
</tr>
<tr>
<td>IT costs</td>
</tr>
<tr>
<td>IT responsiveness</td>
</tr>
<tr>
<td>Risk management</td>
</tr>
</tbody>
</table>

| Business-related benefits  | |
| Shared business platforms | Through data and process standardization, greater data sharing and integrated process standards emerge. |
| Managerial satisfaction   | Although subjective, satisfaction indicates the confidence of business executives in the ability of IT to deliver business value; an increase in senior management satisfaction and business unit IT leadership results from effective EA. |
| Strategic business impacts| EA enables: |
|                          | • Operational excellence |
|                          | • Customer intimacy |
|                          | • Product leadership |
|                          | • Strategic agility |

25.4 EA Frameworks

A number of EA frameworks emerged over the past few decades (Schekkerman 2006), and several studies of evaluations and analyses of select EA frameworks exist, based usually on varying foci (Leist and Zellner 2006; Tang et al. 2004). To provide an overview of typical frameworks, we describe well-known and widely used ones in the following section. Common to these frameworks is reducing enterprise complexities by considering disparate viewpoints and organizing various aspects in ways that make an enterprise understandable. Another important purpose is support for an architecture development process. The EA process provides directions and guidelines to architects to design a model or even an entire architecture.

25.4.1 Zachman EA Framework

Seminal work on the Zachman framework (Zachman 1987) was a key contribution to EA. As IT became more complex, a need arose for a method to organize complex IS. For this reason, primitive
interrogatives of what, how, when, who, where, and why identify aspects of an enterprise. The reification or transformation of an abstract idea into an instantiation is labeled Identification, Definition, Representation, Specification, Configuration, and Instantiation. Intersections among the interrogatives and transformations result in a 6 by 5 matrix. Another important aspect of this framework is consideration of roles (i.e., stakeholders) related to the various perspectives.

The Zachman framework provides a schema for describing the structure of an enterprise; it represents ontology since it provides a structured set of an object’s essential components. It is a meta-model, without implying how to establish architecture. In its current version 3.0, the Zachman framework is an 8 by 8 matrix, depicted in Figure 25.2. It is important to note that this framework was not conceived originally as a methodology since it did not provide a process to instantiate an object (i.e., implement the enterprise). The limited focus on the meta-model design may not warrant consistency among layers and views. Nevertheless, there exist some methodological approaches for applying this framework (e.g., Marques Pereira et al. 2004).

### 25.4.2 Generalized Enterprise Reference Architecture and Methodology

To provide a conceptual basis for IS architectures, the Generalized Enterprise Reference Architecture and Methodology (GERAM) framework was developed by the IFAC/IFIP (IFIP IFAC Task Force on Architectures for Enterprise Integration 2003), building on (Bernus et al. 1996, 1997). One advantage of using GERAM is that it is not a domain-specific architecture; it is a general framework of concepts for designing and maintaining any type of enterprise. GERAM is a framework in which all methods, models, and tools necessary to build and maintain an integrated enterprise are defined; however, it is limited in providing guidelines or reference models for EA in practice. The framework applies to all types of enterprises, including part of an organization, a single organization, or a network of organizations.

Illustrated in Figure 25.3, GERAM is based on the lifecycle concept and identifies three dimensions for defining the scope and content of enterprise modeling:

- **Lifecycle dimensions**, which provide for a controlled modeling process of enterprise entities according to lifecycle activities
- **Genericity dimensions**, which provide for a controlled instantiation process from generic and partial to particular
- **View dimensions**, which provide for a controlled visualization of views of the enterprise entity

To facilitate efficient modeling by enhancing the modeling process, GERAM uses one important concept; it provides various partial enterprise models with an underlying generic enterprise modeling concept. Instead of developing new models within the process, models and concepts are adopted and configured for each situation. Since these concepts and models play an important role in integrating heterogeneous systems, we focus on partial enterprise models (i.e., reference models), including their supported generic modeling concepts. Partial enterprise models are reusable models of human roles, processes, and technologies that capture characteristics common to many enterprises within and across one or more industries. They include various enterprise entities such as products, projects, and companies and may represent each from various viewpoints. To describe multiple levels of abstraction along lifecycle phases, partial models exist at various levels, including concept, requirements, and design levels. Generic enterprise modeling concepts define enterprise modeling constructs. These concepts may be defined using natural language (e.g., glossaries), meta-models (e.g., entity relationship meta-schema), or ontological theories. Bernus et al. (2010) took the GERAM meta-model to the next level by incorporating standards such as IEEE Computer Society (2007) and ISO/IEC (2000), though one of the older frameworks never found a place in practice because of its overwhelming complexity.
### Inventory
- **What**
  - What
  - How
  - Where
  - Who
  - When
  - Why

### Classification names
- **Executive perspective**
  - Inventory identification
- **Business Mgmt perspective**
  - Inventory definition
- **Architect perspective**
  - Inventory representation
- **Engineer perspective**
  - Inventory specification
- **Technician perspective**
  - Inventory configuration
- **Enterprise perspective**
  - Inventory instantiation
  - Process instantiation
  - Distribution instantiation

### Model names
- **Execution context**
- **Business concept**
- **System logic**
- **Technology physics**
- **Tool components**
- **Operation instances**

### Audience perspectives
- **Executive perspective**
- **Business Mgmt perspective**
- **Architect perspective**
- **Engineer perspective**
- **Technician perspective**
- **Enterprise context**

### Scope contexts
- **Business concepts**
- **System logic**
- **Technology physics**
- **Tool components**
- **Operation instances**

---

**FIGURE 25.2** Zachman framework version 3.0.
25.4.3 ARIS

The ARIS framework is a prominent architectural framework often used in practice that represents various aspects of information system design (Scheer 1992). The framework has its roots in the 1980s, with a trend toward corporate data models and material requirements planning. The ARIS house (Figure 25.4) is an architecture that describes business processes, providing modeling methods and meta-structures combined with suggestions for modeling techniques. As a modeling technique and methodology, ARIS is a model for comprehensive, computer-aided business process management, although only representing selected EA views and layers. The concept is also incorporated in a toolset...
to support modeling. To reduce complexity, the model is divided into individual views that represent discrete design aspects:

- **Data view**: Conditions and events form the data view. Events such as order, order reception, and production release are information objects represented by data. Reference field conditions such as customer status and article status are also represented by data
- **Function view**: Functions to be performed and relationships among them
- **Organization view**: Structure and relationships among staff members and organizational units
- **Control view**: An essential component that restores and retains relationships among the various views. By structuring a process, it incorporates a data, function, and organizational view

### 25.4.4 TOGAF

TOGAF (The Open Group 2011) is a comprehensive methodology, including a set of tools to develop an EA. It is based on an iterative process model supported by best practices and a reusable set of existing architecture assets. TOGAF offers clear definitions for various terms, including basics such as architecture(s), views, and stakeholders. TOGAF is organized into three primary sections: Architecture Development Method (ADM), Enterprise Continuum, and Architecture Repository. Additionally, TOGAF provides a comprehensive content meta-model that describes entities and their relationships to ensure common understanding of framework mechanics.

The ADM is a generic method used to realize an organization-specific EA in accord with business requirements. It is organized into several phases, each of which contains various steps to achieve the desired output at the current stage of development. Combined, all phases mark the complete development lifecycle. The **Enterprise Continuum** represents a view of the repository containing all architecture assets, including architecture descriptions, models, building blocks, patterns, and viewpoints, in addition to other architecture and solution artifacts. Architecture and solution artifacts can be internal or external. Internal artifacts are extant architecture assets such as models that are ready for reuse. External artifacts
Enterprise Architecture

comprise industry reference models and architecture patterns that are applicable to aspects of IT and industry and are constantly emerging and evolving. The Architecture Repository supports the Enterprise Continuum by storing classes of architectural output at varying levels of abstraction. The ADM generates this output during several phases. The ADM and its phases are depicted in Figure 25.5.

25.4.5 Other Frameworks

While there exist a number of dedicated EA frameworks (Schekkerman 2006), there are also some related IT frameworks, occasionally working with them, such as IT Infrastructure Library (ITIL) (Office of Government Commerce 2007) and Control Objectives for Information and Related Technology (COBIT) (IT Governance Institute 2007). ITIL is the standard for IT service management. Control objectives for Information and Related Technology (COBIT) began as an IT governance framework, but it has evolved since to a comprehensive IT framework comprised of control objectives, reference processes, management guidelines, and maturity models. Both frameworks mention EA explicitly and demonstrate how they consider EA’s position. COBIT considers EA an explicit IT function, concerned with IT processes with goals derived from business goals as input. ITIL follows a more traditional domain or layer approach, introducing domains such as service, environmental, management, and product architectures.
25.5 EA Modeling

EA modeling is an essential part of contemporary EA practice. Since the EA discipline is concerned with developing a blueprint for an enterprise, practitioners use a variety of models such as data, business interaction, and organizational models (Olivé 2007). An example of an EA modeling language is ArchiMate (The Open Group 2012), based loosely on Unified Modeling Language (UML). ArchiMate supports three layers for modeling—business, application, and technology—each of which contains various modeling elements. The language aligns fully with TOGAF, and each layer connects to ADM phases. Looking at the content meta-model of ArchiMate (The Open Group 2012), each layer offers services. The technology layer offers infrastructure services to the application layer, which in turn offers application services to the business layer. The latter provides the actual business service, consumed either within the enterprise or by external clients. We reflect the SOA paradigm in this language.

Chen et al. (2009) present a domain-modeling approach based on the Zachman framework. UML is one of the most important languages in enterprise modeling. With its great diversity of diagrams, it offers views and models for many situations both at a high level of abstraction when it comes to conceptualization and at a low level for comprehensive technical documentation of inner application aspects. Although the primary foci and applications of UML are software architectures, many enterprise architects use UML (or some derivative) to accomplish the modeling part of their work (i.e., to produce EA artifacts based on models) (Kruchten et al. 2006).

At the center of BAs are business processes, which capture dynamic aspects of the business element in an enterprise. Regarding modeling, there are many process and workflow modeling languages such as Business Process Modeling Notation (Business Process Management Initiative 2011), Event Driven Process Chains (Scheer 1999), Business Process Execution Language (OASIS 2007), and others (List and Korherr 2006). Business Process Management System is a generic software system driven by explicit process designs to enact and manage operational business processes (Weske 2007). It offers a platform that allows both modeling and execution, and is an important part of enterprise modeling and, hence, EA (Grunninger 2003; Scheer et al. 2000). Regardless of models or modeling languages, it is critical that stakeholders understand them (Kaisler et al. 2005).

25.6 Practical Aspects of EA

In larger enterprises, a team of enterprise architects usually conducts the EA function (Murer et al. 2011). In smaller companies, EA is often part of the IT department, with no separate EA unit. Most companies use some of the common EA frameworks even though they are, in many cases, adapted or customized. However, employing a framework alone does not provide an EA; architects are a necessary component. Depending on the company, there are a few architecture job descriptions with varying skill sets such as solution, data, and enterprise architects. We must be concise about who participates in the EA function and what their function is. The Open Group (2011) details a presentation of EA roles and associated skills, and Ahlemann et al. (2012a) outline another explanation of what types of architects are needed. Another practical aspect is EA maturity, indicating how much the function has evolved in terms of various aspects such as architecture planning, value, and stakeholder communication. All EA processes should be supported by a variety of EA tools, and we examine that support in terms of EA processes and services.

25.6.1 EA Maturity

Maturity is a measure that describes advancement and progress in business processes or EA. At each stage, maturity indicates whether requirements are met. Ross (2003) outlines a general description of EA maturity and its stages. The four stages are characterized by management practices to design and protect the architecture. In the first stage, we find business silos, and within them, business cases and project methodologies. In stage two, standardized technologies and structures are in place, including
an IT steering committee, a formal compliance process, centralized handling of enterprise applications, and architects on a project team, among other practices. An optimized core marks the third stage where process owners, EA principles, business leadership of project teams, and IT program managers are in place. The last maturity stage is called business modularity, characterized by an EA graphic, post-implementation assessments, technology research, adoption processes, and a full-time EA team.

There exists a variety of EA maturity frameworks, and Meyer et al. (2011) present an overview and analysis. All of these frameworks possess classifications and characteristics of an EA and a method to assess it. The IT Capability Maturity Framework (IT-CMF) (Curley 2006, 2009) offers five maturity levels for nine capability building blocks. It comprises aspects of planning, practices in the form of governance, processes and value contributions, and organizational and communication aspects. IT-CMF offers detailed assessment for each capability building block.

25.6.2 EA Tools

Companies have broad choices when using EA tools, and these tools help enterprise architects and other stakeholders plan, model, develop, and monitor architectures. As with most business software, companies are not always dependent on vendor solutions, and may choose to develop their own. Most EA tools offer

- EA planning
- EA modeling
- EA simulation and improvement
- EA monitoring and reporting
- Shared artifact repository
- Reusability support

An overview of EA tools and a selection guide is available online from the Institute For Enterprise Architecture Developments (2012), and Wilson and Short (2011) provides a magic quadrant for EA tools.

25.7 Critical Aspects

EA should reduce structural and operational complexity within an organization. Conducted properly, EA delivers projected benefits as discussed earlier, but it can be an investment without any significant return. One challenge with EA is that it is not always obvious what enterprise architects do, how architects generate business value, and whether investments in EA are necessary (Meyer et al. 2012; Robertson 2008). A first step is to generate an EA service model. Mentioned earlier, EA has come a long way over the past few decades, but there is still no single definition or unifying framework. GERAM provides a useful meta-framework, however, with less practical applicability. The Zachman framework and ARIS offers great flexibility and is often used in practice, but is limited with its meta-model and methodology. TOGAF together with ArchiMate supports modeling in conjunction with an ADM but can be challenging to adapt in a practical setting. In practice, often a combination of several frameworks can be found. Also, several studies of evaluations and analyses of select EA frameworks exist (e.g., Leist and Zellner 2006; Tang et al. 2004). Furthermore, distinctions among processes such as EA development and EA management do not appear in the literature. Although Schönherr (2008) points out that common terminology is important for the EA discipline, we are unable to offer concise descriptions of what EA is and does, and Kaisler et al. (2005) mention critical problems with EA that are barely solved after years of research.

25.8 Summary

Since its origins, described as IS Architecture in the 1980s and 1990s, EA evolved into an extensive domain with numerous approaches and frameworks. A core characteristic of EA is that it enables and supports constant transformations of an enterprise from a current to a target state. EA supports this
transformation with tools and frameworks to manage complex enterprise systems. We outline prominent frameworks and modeling approaches and describe the importance of practical tools. EA provides directions and guidelines for high-level enterprise evolution by encapsulating and relating business aspects, application portfolios, information and data, and technology. Related to strategic management, alignment of business and IT is one of the most important drivers of EA. An effective EA function yields a number of benefits such as reduced complexity, reduced risks, reduced costs, and improved business–IT alignment (Lankhorst 2009; Op ’t Land et al. 2009; Ross et al. 2005, 2006; van Steenbergen et al. 2011). Enterprises evolve constantly; producing drivers for employing an EA function—with all its capabilities—make it an ever-emerging discipline. New developments and innovative modeling approaches, tools, and model-based EA frameworks can be expected, combined with practical methods to employ and value EA in enterprises. These are the foci of current research and development.

References


Henderson, J. C. and Venkatraman, H. Strategic alignment: Leveraging information technology for transforming organizations, IBM Systems Journal (38), 1993, pp. 472–484.
25-15


