Chapter 15
Frameworks and Reference Architectures

Within the field of Enterprise Modeling, substantial work has been spent on defining frameworks and architectures. In comparison to EM methods (see Chap. 14), frameworks and architectures do not focus on procedures for the actual modeling process, notations, or modeling languages, but they address the modeling domain or the results of the modeling process.

Most frameworks were developed within a specific application domain or for an enterprise function and structure this domain and function. Typical organizational structures and process areas, important concepts, or building blocks of enterprises or solution are identified and described, documented, and made available as templates or generic models. The intention is to make those elements of previous modeling projects available and reusable, which are not specific for a single enterprise, but general for a whole industry domain or for an enterprise function.

Architectures typically focus on building blocks and their relationships, while frameworks can have different ambitions, like combining architectures with design or modeling procedures or reusable models or model fragments with the way of developing them into a specific solution. Both approaches have in common that they have to be adapted for the modeling project or the enterprise under consideration, i.e., they are rather a rough blueprint than a ready-made solution.

Frameworks and architectures are part of this book, since they might be useful in different phases of an EM project. Depending on the intention and the purpose of the project, frameworks and architectures could be used:

- In the scoping phase to help structuring the overall application domain and identify, together with the stakeholders in the enterprise, which areas to focus on
- When designing the "to be" situation to identify potential best practices in the domain or to get inspiration for how to structure the overall field
- In the analysis phase to reuse proven definitions of general concepts in the domain, as long as they fit to the understanding in the enterprise under consideration.

K. Sandkuhl et al., *Enterprise Modeling*, The Enterprise Engineering Series,
DOI 10.1007/978-3-662-43725-4_15

It should be noted that identifying and understanding the problem to be solved by an EM project usually is not supported by frameworks and architectures, i.e., understanding problems and issues to be considered is prime and should come first. Selecting and using frameworks or architectures from the domain can be recommendable, but should be the second step.

From the many existing frameworks and architectures, only a few will be discussed in this section:

- *The Zachman framework*—a framework providing support to analyze enterprises which during many years was very popular in industry,
- *GERAM*—an approach which combines different industrial reference architectures to a generalized framework,
- *TOGAF*—the open group architecture framework.

We will give TOGAF more room in this chapter than Zachman's framework and GERAM, since the interest in TOGAF in industry, the public sector, and academia seems to be significantly higher than for the other two approaches. Many more exist and are covered in the literature, like, e.g., DoDAF (Department of Defense 2007) or MODAF (Ministry of Defense 2008).

15.1 The Zachman Framework

The Zachman Framework for enterprise architecture was proposed by John A. Zachman, a former IBM employee, who developed the framework as tool to help analyze enterprises or parts of enterprises. The core idea of the analysis tool is that the complexity of an enterprise can be more easily analyzed and captured if you use the different perspectives of the stakeholders interested in the analysis result and if you classify the enterprise information according to the content of the analysis subject. This idea resulted in a two-dimensional matrix consisting of perspectives and subject classifications, called "abstractions", which can be used to guide the analysis work. Furthermore, the matrix also provides a classification scheme for what Zachman calls "descriptive representations of an enterprise" (Zachman 1987).

The perspectives distinguished in Zachman's framework are depicted in Fig. 15.1:

- *The owner's perspective*. The owner in this context either is the owner of the enterprise under consideration or the recipient of the end product produced. The latter is relevant if an organization form is considered where many partners jointly produce a complex product, like a consortium of partners producing a ship.
- *The designer's perspective*. The "descriptive representations" in this perspective are meant to form the basis for implementing what is desired by the owner, i.e., the intermediary between owner's and builder's perspective. In an enterprise,

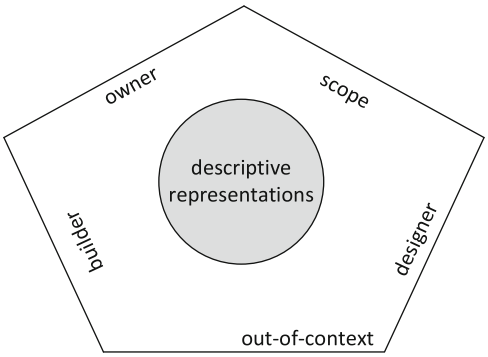

Fig. 15.1 Descriptive representations of Zachman's framework according to (Zachman 2003)

Structure	Process	Flow	People	Events	Motivation
What is it comprised of?	How does it function?	Where is it located?	Who is responsible for operations?	When does it happen?	Why does it happen?

Fig. 15.2 Abstractions of Zachman's framework according to (Zachman 2003)

this usually is the systematic and conceptual design of the administrative, manufacturing, and management processes and structures.
- *The builder's perspective*. This perspective captures how the design perspective actually is to be implemented, which takes into account existing technical or organizational constraints.
- *The scope perspective* establishes the inner and outer limits of what has to be considered in the other perspectives, i.e., what has to be subject of the descriptive representations and what is beyond the limit.
- *The out-of-context perspective* captures aspects out of the context of the enterprise or product modeled but still in the scope of the modeling project. In an enterprise, this could be the actual physical products manufactured, in contrast to the design and manufacturing processes and blueprints of these products which would be subject of the designer's and builder's perspective.

With the above perspectives as one dimension, the "abstractions" are the second dimension of the framework (see Fig. 15.2) and structure the different characteristics and aspects required to describe the subject under consideration:

- What the subject or object under consideration is comprised of
- How it works, i.e., the specification of a process or of the functionality
- Where the subject or its components are located, i.e., the spatial dimension
- Who is responsible for what and who performs which work
- When activities or events happen in relation to each other
- Why things happen or are performed in the enterprise. This motivational dimension usually relates to the strategy

Zachman describes the perspectives and abstractions as primitive and comprehensive in the sense that each abstraction and each perspective is different from each of the other ones and that no other perspectives and abstractions are needed to provide a complete knowledge base about the enterprise. For the single cells of the matrix generic models are available which have to be specialized for the enterprise under consideration. If different levels of detail are required, they have to be modeled within the different cells of the matrix, i.e., the columns are not intended to and do not provide a possibility to model different levels of detail.

In Enterprise Modeling, the Zachman framework can be used as a general guideline what to consider in order to not forget certain aspects. However, it implies an ambition to reach a certain level of completeness, which is not required for all EM purposes.

The framework has been very popular and widely used in the first decade of the 2000s. Judging from the decreasing number of publications and experience reports, it nowadays is no longer so widely used. One reason for this might be that Zachman's framework formed the basis for the *Technical Architecture Framework for Information Management (TAFIM)* developed by the US department of defense in 1994. TAFIM later formed one of the starting points for TOGAF (see Sect. 15.3) which now is widely used in industry and which somehow replaced Zachman's framework.

Further readings regarding the Zachman Framework is available on the Web site http://www.zachmaninternational.com.

15.2 GERAM

The IFAC/IFIP task force on architectures for enterprise integration produced in 1998 the Generalized Enterprise Reference Architecture and Methodology (GERAM) (Williams 1995). GERAM was based on a number of previous developments which complemented each other:

- CIMOSA (Computer Integrated Manufacturing Open System Architecture) has been developed in a European project and aims to support the enterprise integration. This architechture is based on the system life cycle concept, and offers a modeling language, methodology, and supporting technology.
- GRAI is a methodology which was developed at the University of Bordeaux in the 1990s. The methodology includes a graphical modeling technique and considers an organization as a complex system consisting of three subsystems: the decision subsystem, the information subsystem and the physical subsystem.
- PERA is a methodology originating from Purdue University (USA) and includes a generic list of tasks in a manufacturing plant and a hierarchical functional framework for relating them to each other.

The intention of GERAM was to combine the above described industrial reference architectures to a generalized framework with all components needed for

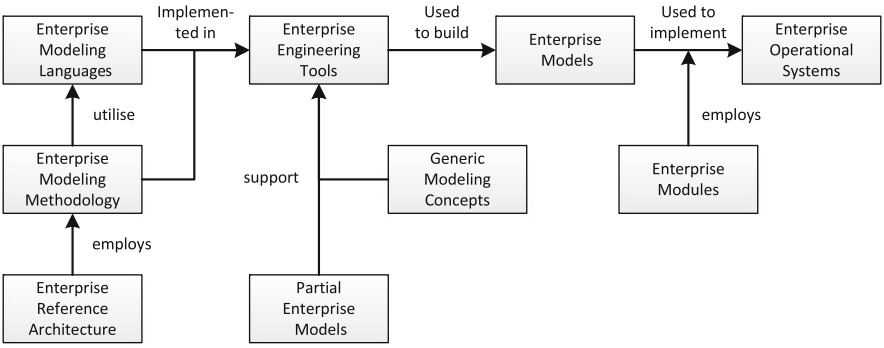

Fig. 15.3 Overview to GERAM components (Lillehagen and Krogstie 2008)

enterprise engineering and enterprise integration. "Enterprise engineering" as a term indicates that GERAM is part of an industrially driven initiative which considers enterprises as complex systems that can be "engineered" in a similar manner as complex products, i.e., for well-known problems or tasks in an enterprise, the processes and structures for solutions to these problems and tasks can be predefined and captured as models. These models and their implementation in real-world systems form "components" representing the knowledge about an enterprise in a certain domain that can be reused if changes in the enterprise or integration with other enterprises need to be implemented.

The GERAM framework consists of eight elements depicted in Fig. 15.3. These elements are:

- *The GERA analysis and modeling framework*: this framework defines three dimensions for identifying scope, subject, and content of Enterprise Modeling: lifecycle, instantiation (level of abstraction), and view. The lifecycle dimension in GERA consists of the phases identification, concept, requirements, (preliminary and detailed) design, implementation, operation, and decommission. The instantiation dimension basically defines different levels of abstraction, which are generic, partial, and particular. The view dimension includes views regarding the purpose of the activity (e.g., customer service, management, and control), the model content (resource, information, organization, function), the physical manifestation (e.g., software, hardware), and the means of implementation (human, machine). When using these dimensions, enterprise modelers and enterprise model users are supported in defining the scope of a modeling project by selecting which aspects of the different dimensions are required, in systematically modeling the defined scope, and in structuring the modeled knowledge about the enterprise.
- *Enterprise Engineering Methodology*: the methodologies provided by GERAM are meant to describe the process to be performed for every aspect of the lifecycle dimension in a generalized way. The methodologies are supposed to be applicable regardless of the industry domain concerned and support their

users in the process of enterprise engineering and integration, both for management-related and engineering-related aspects.

- *Enterprise Modeling Languages:* modeling languages define the constructs and the notation to be used for expressing enterprise models (see also Sect. 3.1)
- *Generic Enterprise Modeling Concepts*: concepts frequently used in Enterprise Modeling, engineering, and integration should be consistent throughout the different activities. In case these concepts are generic, i.e., not specific for a certain enterprise only, they should be defined once in order to allow for reuse. Potential ways to define generic EM concepts are glossaries, meta-models, and ontologies. GERAM recommends to use meta-models for generic concept definition.
- *Partial Enterprise Models* capture concepts of certain aspects of GERA that are common to many enterprises. These partial models can be considered as reusable parts or even reference architecture which can speed up the modeling process by reusing these proven models if suitable for the enterprise under consideration.
- *Enterprise Engineering Tools* support the process of the different activities in enterprise engineering and integration, i.e., analysis, design, reuse, and use of enterprise models.
- *Enterprise Modules are building blocks or systems* implemented in an enterprise, which can be accessed and utilized in an enterprise or offered as resources on the market. Often, these enterprise modules are implementations of partial enterprise models.
- *Enterprise Models* capture selected aspects of an enterprise in a model (see also Sect. 3.1) in a defined EM language.
- *Enterprise Operational Systems* usually consist of the hardware and software required for operations in a particular enterprise, i.e., they are platform supporting operations in an enterprise.

15.3 TOGAF

The purpose of this section is to give a brief view on TOGAF 9.1, an architecture framework, which originally established in the 1990s and since then evolved as a leading standard for developing an Enterprise Architecture (EA). More than 100,000 downloads, 16,000 certified practitioners, 220 corporate members are in touch already with the TOGAF Framework since 2011 (Weismann 2011).

The section is structured as follows: Section 15.3.1 presents briefly the relationship between Enterprise Modelling and EA and describes what TOGAF is, where it has its origins, and how it evolved over time. Section 15.3.2 focuses on EA and on how TOGAF interprets and defines this term. The main components of TOGAF are presented in Sect. 15.3.3 with focus on the Architecture Development Method (ADM) and the Enterprise Continuum. Finally, a summary of the main characteristics of TOGAF is included.

15.3.1 Enterprise Modeling and TOGAF

Business and IT stakeholders in a company have different views of the enterprise and, therefore, different viewpoints on its architecture (Glissmann and Sanz 2011). Consequently, special techniques for describing enterprise architectures (EA) in a coherent way and communicating them with all relevant stakeholders are necessary in order to create an integrated perspective of the enterprise. EA models support bridging the communication gap between enterprise or IT architects and stakeholders from business [5]. In general specific EA models are used to map architecture descriptions that represent different and/or partial views of the whole EA. For instance, 4EM can be used just for specifying important goal components to get an overview about the enterprise strategy as well as 4EM can be used to model problem issues like described in the case study (see Chap. 6).

TOGAF stands for The Open Group Architecture Framework and presents a "comprehensive architecture framework and methodology, which enables the design, evaluation, and implementation of the right architecture for an enterprise" (The Open Group 2011). It provides methods, tools, and best practices to support the "acceptance, production, use, and maintenance of an enterprise architecture" (The Open Group 2011), which can be customized to and implemented in different companies for their needs.

The original version of TOGAF, Version 1, was introduced in 1995 by the US Department of Defense Technical Architecture Framework for Information Management (TAFIM). After that the Department of Defense gave The Open Group the permission to take over the further development of the framework. Since then, more than 300 member organizations of The OpenGroup's Architecture Forum are constantly working on TOGAF, adding new features and concepts.

TOGAF did not always focus on EA. Initially, it included only technical architectures (Version 1–7). With the release of Version 8, called Enterprise Edition, it also began to cover the business architecture domain. The latest version, TOGAF 9.1, was launched in December 2011. All related documentation about TOGAF can be obtained from The Open Group Web site, so that the usage is encouraged (Harrison and Varveris 2004).

TOGAF has two main components. The first core component is the Architecture Development Method, or the ADM for short, which defines iterative processes for developing and maintaining an organization's enterprise architecture. The Enterprise Continuum is the second core component to TOGAF. It describes a collection of reusable assets, called building blocks, which supply architects with reference architectures, models, and processes, which can be adopted to create new architectures (Temnenco 2007).

15.3.2 Enterprise Architecture Management in TOGAF

Before characterizing the four types of architectures, which TOGAF deals with, it is essential to define how to understand the architecture concept. "An architecture is a fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principle guiding the organization's design and evolution" (Lankhorst et al. 2009). Lankhorst describes architecture in an analogy:

> *"Suppose you contract an architect to design your house. You discuss how rooms, staircases, windows, bathrooms, balconies, doors, a roof etc. will be put together. You agree on a master plan, on the basis of which the architect will produce detailed specifications, to be used by the engineers and builders. How is it that you can communicate so efficiently about that master plan? We think it is because you share a common frame of reference: you both know what a "room" is, a "balcony", a "staircase" etc. You know their function and their relation. A "room" for example serves as a shelter and is connected to another "room" via a "door". You both use mentally an architectural model of a house"* (Lankhorst et al. 2009).

No common definition of the term EA has emerged yet. It depends on the different point of views. Thus it is mainly related to IT Architecture focusing on creating and using IT systems as well as Business Architecture, which concentrates on achieving the business strategy by specific, suitable actions (Aier et al. 2008). EA is the idea of modeling the elements, roles, responsibilities, and systems, as part of the enterprise infrastructure, and their relations. In this sense the capture of all behavior that goes on in an organization including processed data, shared tasks such as who does what, why everything is done, and where everything is (Harrison and Varveris 2004). It is a coherent whole of principles, methods, and models that are used in the design and the realization of the enterprise organizational structure (Lankhorst et al. 2009).

There are four architecture domains that are accepted as subsets of an EA (Fig. 15.4) and TOGAF is designed to support all of them.

The Business Architecture defines the business strategy, governance, organization, and key business processes. This architecture addresses the concerns of the users, planners, and business managers (Glissmann and Sanz 2011). The Data Architecture describes the structure of an organization's logical and physical data assets and data management resources . Its objective is to define the major types of data, necessary to support the business. This architecture addresses the concerns of database designers and administrators (Glissmann and Sanz 2011). In some organizations, Data Architecture is also called Information Architecture. The Application Architecture provides a blueprint for the individual application systems to be deployed, for their interactions and their relationships to the core business processes

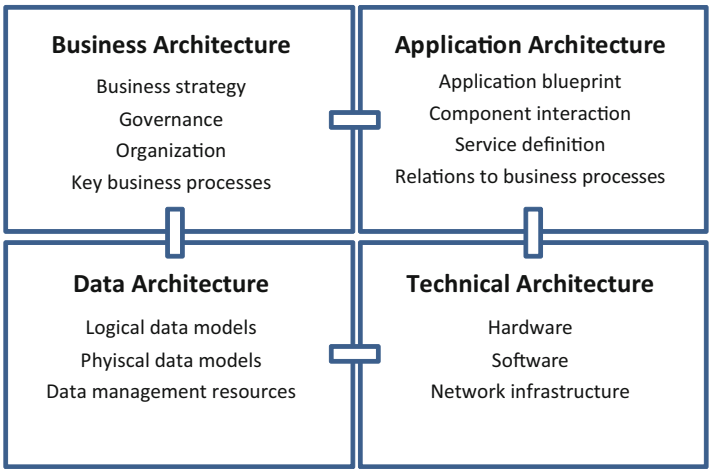

Fig. 15.4 TOGAF architecture domains

Table 15.1 Summarized architecture types supported by TOGAF (The Open Group 2011)

Architecture type	Description
Business architecture	The business strategy, governance, organization, and key business processes
Data architecture	The structure of an organization's logical and physical data assets and data management resources
Application architecture	A blueprint for the individual applications to be deployed, their interactions, and their relationships to the core business processes of the organization
Technology architecture	The logical software and hardware capabilities that are required to support the deployment of business, data, and application services. This includes IT infrastructure, middleware, networks, communications, processing, and standards

of an organization. The Technology Architecture describes the physical realization of an architectural solution. The logical software and hardware capabilities, which are required to support the deployment of business, data, and application services, are also defined in this dimension. The four architechture types are summarized in Table 15.1.

Those four dimensions are intimately connected through the relationships between the individual meta-model elements. For instance, a data entity (DA) is used by a logical application component (AA), which is used by an actor in a business process to meet business objectives (BA). TA supports the application component (Glissmann and Sanz 2011).

The idea of Enterprise Architecture Management (EAM) includes the planning, transforming, monitoring, and improvement of the different architecture levels. In this context, the Enterprise Architecture (EA) serves as map with information of the

current situation of its elements and dependencies. There is a variety of reasons for implementing EAM:

- It supports delivery of the business strategy
- It facilitates management and exploitation of information is key to business success and competitive advantage
- It facilitates management of stakeholder concerns that needed to be addressed by IT systems
- It enables management of complexity and changes to business/IT
- It enables the right balance between IT efficiency and business innovation
- It improves transparency and manage risks

TOGAF also helps implementing a strategy oriented control of the different architectural levels, which enables economic success.

15.3.3 Components of TOGAF 9.1

TOGAF 9.1 consists of seven parts presented in the following parts of this section.

15.3.3.1 Part 1: Introduction

The first part of the TOGAF framework involves the introduction the EA key concepts. Therefore, it contains the definitions of terms used throughout TOGAF and release notes detailing the changes between the different versions of TOGAF. Questions like "What is an enterprise? Why do I need an enterprise architecture? Why do I need TOGAF as a framework for enterprise architecture?" will be answered in this section.

15.3.3.2 Part 2: The Architecture Development Method

The architecture development method (ADM) is a step-by-step approach to develop and use an EA. The main purpose of the approach is to help to derive a specific architecture from a set of common architectures to meet the business requirements of an enterprise (Josey et al. 2009). The ADM supports the development of an architecture in the four different domains (business, application, data, and technology), described in the previous section. It consists of ten consecutive phases (see Fig. 15.5) enclosed in a loop (Temnenco 2007).

Each phase has an input, an output, which at the same time serves as an input for the next following phase, and a number of steps. In the following these phases will be described briefly:

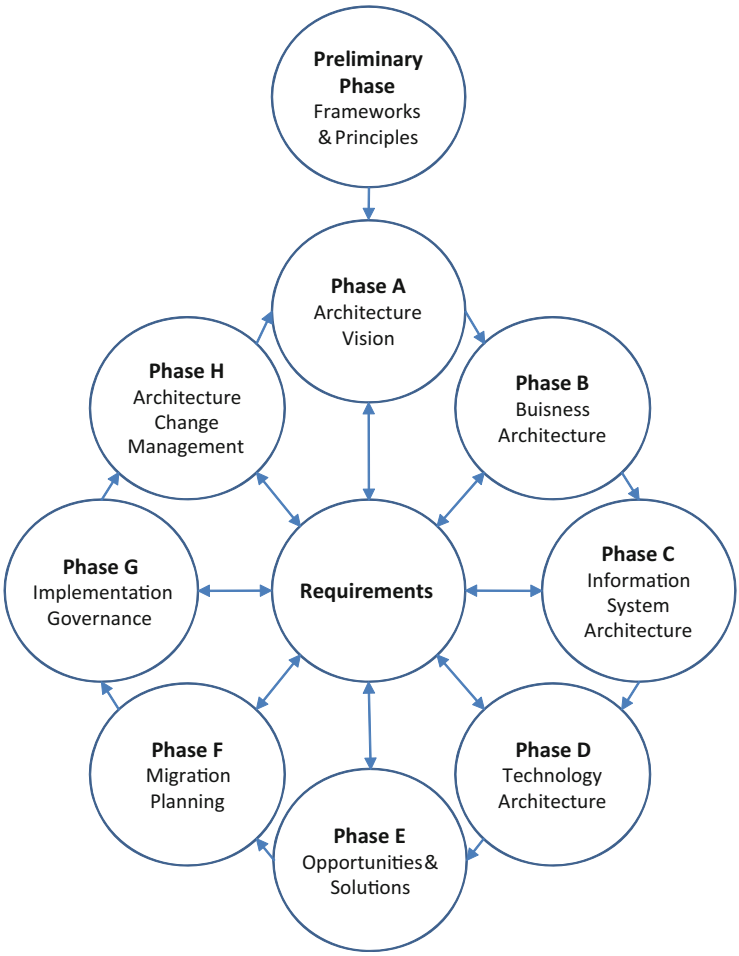

Fig. 15.5 The architecture development method cycle (The Open Group 2011)

- *Preliminary Phase:* The main goal of this phase is to prepare the organization and get the stakeholders ready for a successful TOGAF project. Typical steps of this phase are defining the scope of the enterprise, establishing the team and the organization, and determining the architecture principles.
- *Phase A:* Architecture Vision: Phase A is dedicated to articulating the EA vision and principles, and presents the initial phase of the architecture development cycle. Its most crucial objectives are to obtain a management commitment for this particular cycle of the ADM and to validate business principles, goals, and key performance indicators. During the preliminary phase and Phase A, it must be clarified, how much information will be captured, how it will be maintained, what notations or methods are used to build the enterprise models.

- *Phase B:* Business Architecture (BA): Phase B describes the current and the target business architecture and tries to determine the gap between these two. The motivation for developing a BA is to support the Architecture Vision which was agreed upon in the previous phase. Adequate tools which can be applied in this phase in order to develop the required models are, e.g., BPMN and UML (Harrison and Varveris 2004).
- *Phase C:* Information Systems Architectures: Phase C focuses on the Information Systems Architectures, which comprise the Data and Application Architectures. These architectures can be developed either sequentially or concurrently (Josey et al. 2009). In this phase, business-supporting data types and sources are to be described in such a way that the stakeholders understand them. Henceforward, the application systems, which can process the data, are to be defined.
- *Phase D:* Technology Architecture: Phase D deals with documenting the organization of the IT Systems, embodied in the enterprise hardware, software, and communication technology. The completion of Phases B and C is a prerequisite for moving on to Phase D. The development of all four architecture domains are covered after the Phases B, C, and D are finished.
- *Phase E:* Opportunities and Solutions: Phase E is the first phase, which is directly concerned with implementation (Josey et al. 2009). It has two main purposes—to clarify the opportunities presented by the target architectures, which have been identified in previous phases, and to outline the potential solutions. The important outputs of this phase are a major implementation project and an updated Application Architecture which can serve as a blueprint to be used by future implementation projects.
- *Phase F:* Migration Planning: The proposed implementation projects need to be prioritized so that a detailed planning can be performed. In this phase the enterprise knows how to move from the baseline to the target architecture by finalizing a detailed Implementation and Migration Plan. The blueprint, developed in the previous phase, is also handed over to the implementation teams.
- *Phase G:* Implementation Governance: In phase G, the projects are started as a planned program of work that is accompanied by implementation process oversights.
- *Phase H:* Architecture Change Management: Phase H provides a change management process to ensure that the designed architecture corresponds to the needs of the enterprise. If the enterprise needs change then these changes will be realized in the architecture in a controlled and procedural manner. Phase H can also result in a request for a new architecture framework and, if so, another cycle of the ADM is initiated.
- *Requirements Management:* The Requirements Management process applies to all phases of the ADM cycle because TOGAF is a requirements-centric approach (Temnenco 2007). Generally, architecture deals with change and uncertainty in requirements, since it bridges the gap between the expectations of the stakeholders and delivered solutions. That is why Requirements Management has a central meaning to TOGAF. This phase defines and stores all types of requirements, and feeds them in and out of the relevant ADM phases (Josey et al. 2009).

15.3.3.3 Part 3: ADM Guidelines and Techniques

This part provides a set of guidelines and techniques to support the application of the ADM. It deals with different scenarios, including different process styles (e.g., the use of iteration) and also specific requirements (e.g., security). The techniques support specific tasks within the ADM (e.g., defining principles, business scenarios, gap analysis, migration planning, risk management, etc.).

According to (The Open Group 2011; Keller 2012) the third part deals with the following issues:

- Using ADM as a cyclic process. This is about managing iteration and the potential strategies for applying iterative concepts to the ADM.
- Applying the ADM across the Architecture Landscape. It is about the different types of architecture engagement that may occur at different levels of the enterprise. It is also about how the ADM process can be focused to support different types of engagement or levels of granularity.
- Doing security engineering while using TOGAF. An overview of specific security issues that should be considered during different phases of the ADM is provided as well as aspects of using TOGAF for SOA support, stakeholder management, or architecture patterns.

15.3.3.4 Part 4: Architecture Content Framework

The Architecture Content Framework provides a detailed structural model for architectural content that allows the major work products including deliverables and artifacts within deliverables that an architect creates to be consistently defined, structured, and presented in Architecture Building Blocks (ABBs). The fourth part supports the following aspects:

- Increasing the consistency in the outputs of TOGAF.
- Providing a comprehensive checklist of architecture outputs.
- Promoting better integration of work products.
- Providing a detailed open standard for how architectures should be described.
- Including a detailed meta-model (see Fig. 15.6)

15.3.3.5 Part 5: Enterprise Continuum and Tools

The Enterprise Continuum (EC) is “a categorization mechanism useful for classifying” all assets relevant to an enterprise. The result of its practical implementation is an Architecture Repository which presents a collection of “reference architectures, models, and patterns that have been accepted for use within the enterprise”

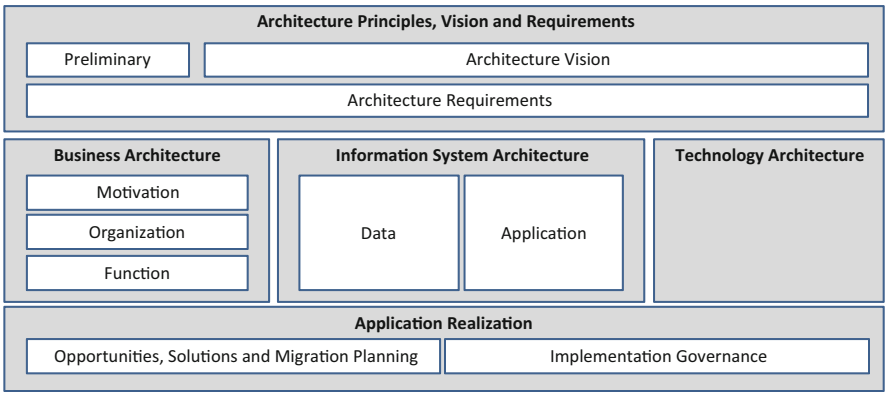

Fig. 15.6 Content meta-model (simplified) (The Open Group 2011)

(The Open Group 2011). The EC focuses strongly on the two ideas of reusability and understandability.

Reusability is achieved through the concept of the building blocks. A building block (BB) is "a (potentially reusable) component of business, IT, or architectural capability that can be combined with other building blocks to deliver architectures and solutions" (The Open Group 2011). The delivered architectures and solutions can then be used in two directions: (1) the general ones can be adapted to fulfill specific needs and (2) the specific ones can be generalized for further reuse.

The idea of understandability is achieved through two concepts. The first one is the concept of sequentially moving from generic to specific, from abstract to concrete, which helps everybody involved in the architecture development process to understand where exactly they are in the continuum and which type of architecture is currently in focus. The second concept is the separation between architectures and solutions—the EC is divided into two continua, the Architecture Continuum and the Solutions Continuum.

The Architecture Continuum (AC) is "a repository of architectural elements with increasing detail and specification" (The Open Group 2011) and has four states: foundation architectures, common systems architectures, industry architectures, and organization-specific architectures.

Foundation Architectures present architectures of building blocks and corresponding standards that support all the Common Systems Architectures and, therefore, the complete enterprise operating environment.

Common Systems Architectures are architectures of particular problem domains within an organization. Examples of such architectures are security architectures, management architectures, network architectures, etc.

Industry Architectures are architectures that integrate common systems with industry-specific components to create solutions to problems within a particular

industry. Such architectures contain industry-specific logical data, industry-specific process models and applications, and industry-specific business rules.

Organization-Specific Architectures contain organization-specific business models, data, applications, and technologies. They reflect requirements and define BB specific to a particular enterprise, and provide the criteria to measure and select appropriate products, solutions, and services.

The Solution Continuum (SC) provides particular solutions to implement the corresponding architectures from the AC.

Therefore, the SC has also four states as the AC: Foundation Solutions, Common Systems Solutions, Industry Solutions, and Organization-specific Solutions.

The solutions are developed with the help of Solution Building Blocks (SBB), which, just as the ABB, increase in detail and specification in each state.

The SBB include concepts, tools, products and services such as programming languages, operating 10 systems, ERP-Systems, IT Organization-Management standards and principles such as ITIL, and others.

15.3.3.6 Part 6: TOGAF Reference Model

Part 6 of the TOGAF Framework provides two reference models:

1. The Foundation Architecture/ Technical Reference Model (TRM)
2. The Integrated Information Infrastructure Model (III-RM).

The foundation architecture provides an architectural approach of generic services and functions which should support building more specific architectures and architectural components. The foundation architecture is embodied within the universally applicable TRM that represents a model and taxonomy of generic platform services (Fig. 15.7).

The III-RM is defined as a subset of the TRM in terms of its overall scope. It supports the design of an integrated information infrastructure by defining a taxonomy concept and associated visual representation of the interrelationship of its components. According to (The Open Group 2011) the main objective of the TOGAF TRM is the allocation of a widely accepted core taxonomy and an appropriated visual representation of them.

15.3.3.7 Part 7: Architecture Capability Framework

This part discusses the organization, processes, skills, roles, and responsibilities required to establish and operate an architecture practice within an enterprise as well as provide guidance on establishing an operational practice. In order to achieve TOGAFs view on successfully operating architecture functions, it is necessary to put in place appropriate organizational structures, processes, roles, responsibilities, and skills to realize the Architecture Capability. In this section of the TOGAF Framework, a set of reference materials for how to establish such an architecture

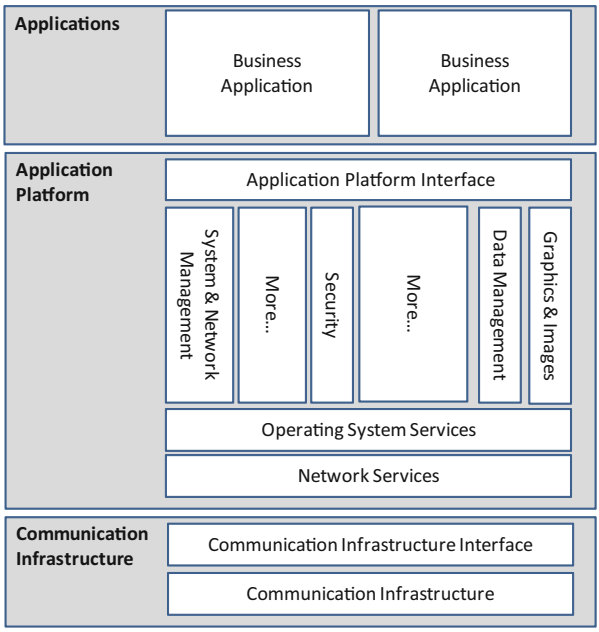

Fig. 15.7 TRM overview (simplified) (The Open Group 2011)

function is presented. The Architecture Capability Framework is not intended to be a comprehensive template for operating an enterprise Architecture Capability; it provides a number of guidelines to support key activities (The Open Group 2011).

15.4 Summary

This session introduced TOGAF—The Open Group Architecture Framework in its latest version, Version 9.1. It gave a brief insight into components of TOGAF where the ADM is the very core of TOGAF. TOGAF covers the whole lifecycle of EA and EAM—from the idea, covered in the Architecture Vision (Phase A), to the control of changes, handled in the Architecture Change Management (Phase H).

In addition to this, through the Requirements Management, new external drives, especially business strategies and requirements, can be added to the architecture during the whole lifecycle. The EC presents a virtual repository of architecture assets which evolve from generic to specific and can be adopted to develop a target architecture by starting with a common architecture and finishing with one, specific to the organization.

TOGAF provides a very interesting approach to EA and EAM which can help solve many different problems in an organization. It has the potential to help organization to deeply examine their organization, to understand how it currently

operates, and to define how it must function. It encourages stakeholders and management to work together in order to achieve the desired organization. However, TOGAF is a very complex framework and its use requires thorough preparation. Only reading its documentation and visiting several workshops will not sufficiently prepare for dealing with TOGAF. This view is also shared in (Sessions 2007). There the author argues that TOGAF focuses on how to develop an EA and not on how to develop a good one. Therefore, the final architecture can either be good, bad, or indifferent. The result completely depends on the knowledge and skills of the TOGAF architects.

In addition, it is recommendable for TOGAF users to find access to communication platforms, where they can exchange knowledge, thoughts, ideas, and experience with others on different EA and TOGAF matters.