

Analyzing The Open Group Architecture Framework from the GERAM Perspective

Pallab Saha*

Institute of Systems Science, National University of Singapore, 25 Heng Mui Keng Terrace, Singapore 119615

Abstract

This paper analyzes The Open Group Architecture Framework (TOGAF) Enterprise Edition and its mapping onto the Generalized Enterprise Reference Architecture and Methodology (GERAM) framework / ISO IS 15704:2000 requirements. The analysis compares and contrasts these frameworks on multiple aspects that include: life cycle phases, temporality and succession, modeling frameworks, modeling languages, methodologies, entity types and reference models. The paper then discusses the role of TOGAF in the context of GERAM compliant enterprise architecture development including suggestions for issues and areas to be addressed.

Keywords: Enterprise Engineering; Enterprise Modeling; Technical Reference Model

1. Introduction

With the advent of globalization and free trade, enterprises are operating as large and complex federation of autonomous business units. In order to continually conduct business in a professional manner enterprises have to design facilities, services and resources in a globally distributed environment. This creates severe pressures on enterprises to avoid duplication and overlaps, providing the need for organizations to scientifically design enterprises and manage them through their entire life cycles. Enterprise architecture (EA) is the discipline of designing enterprises guided with principles, frameworks, methodologies, requirements, tools, reference models and standards. EA is a set of descriptive representations that are relevant for describing an enterprise such that it can be produced to management's requirements and maintained over its period of useful life [20, 21]. This is necessitated by the fact that most organizations have grown organically where enterprise design is based largely on intuition, rather than well defined and executed principles [3]. While components of enterprise architecture involve several areas of focus¹ information technology (IT) architecture plays a crucial role in today's enterprises given the increasing dependency shown by organizations on IT. Of the

¹ Enterprise architecture is typically categorized into: (1) Business Process Architecture, (2) Information Architecture, (3) Technology Architecture and (4) Applications Architecture.

several frameworks currently available [3, 6, 7, 8, 10, 11, 14, 18], this paper takes into consideration one the most prevalent architectural frameworks² in the industry and analyzes the same against the GERAM / ISO IS 15704:2000 requirements. The objective of this detailed analysis is to provide prospective users of TOGAF an understanding of GERAM / ISO IS 15704: 2000 requirements, and the extent to which TOGAF meets these requirements. While analysis of several other frameworks against the GERAM / ISO IS 15704: 2000 requirements does exist [12], currently there is no mapping between TOGAF and GERAM and the paper attempts to address this gap.

1.1. A Short Overview of GERAM

The overriding goal of the Generalized Enterprise Reference Architecture and Methodology (GERAM) is to encompass and generalize the commonalities of various existing EA frameworks and EA reference architectures [2, 9]. The GERAM is not yet another EA framework or EA reference architecture. GERAM aims to classify prevalent EA frameworks and their associated artifact types (methodologies, reference models, ontologies etc.). The GERAM consists of several components as depicted in Figure 1.

The scope of GERAM encompasses all knowledge required for enterprise engineering³ and enterprise integration⁴ with the intention of unifying several disciplines such as industrial engineering, management science, control engineering and information and communication technology to build a coherent organizational design [3]. The GERAM V1.6.3 provides description of all the components shown in Figure 1 thereby providing standards for other frameworks to follow and aspire for. GERAM however, is not prescriptive about any particular set of tools, methodologies, templates and models, but specifies the criteria to be satisfied by any set of selected tools and methods to be compliant. The central component in the GERAM framework is Generalized Enterprise Reference Architecture (GERA) which specifies the basic / core concepts to be utilized in an EA initiative. Besides GERA, GERAM identifies requirements regarding the process methodology, modeling languages, tools and enterprise models necessary for enterprise engineering as other components. The GERAM framework components would form the basis for mapping and analysis of TOGAF in this paper.

² The Open Group Architecture Framework (TOGAF) Version 8.1 Enterprise Edition (December 2003).

³ *Enterprise Engineering* is the discipline by which organizations can design, deploy and maintain an integrated state of the enterprise throughout its lifecycle.

⁴ *Enterprise Integration* aims to eliminate organizational barriers in order to improve interoperability thus creating synergies within the enterprise to enhance efficient and adaptive business operations.

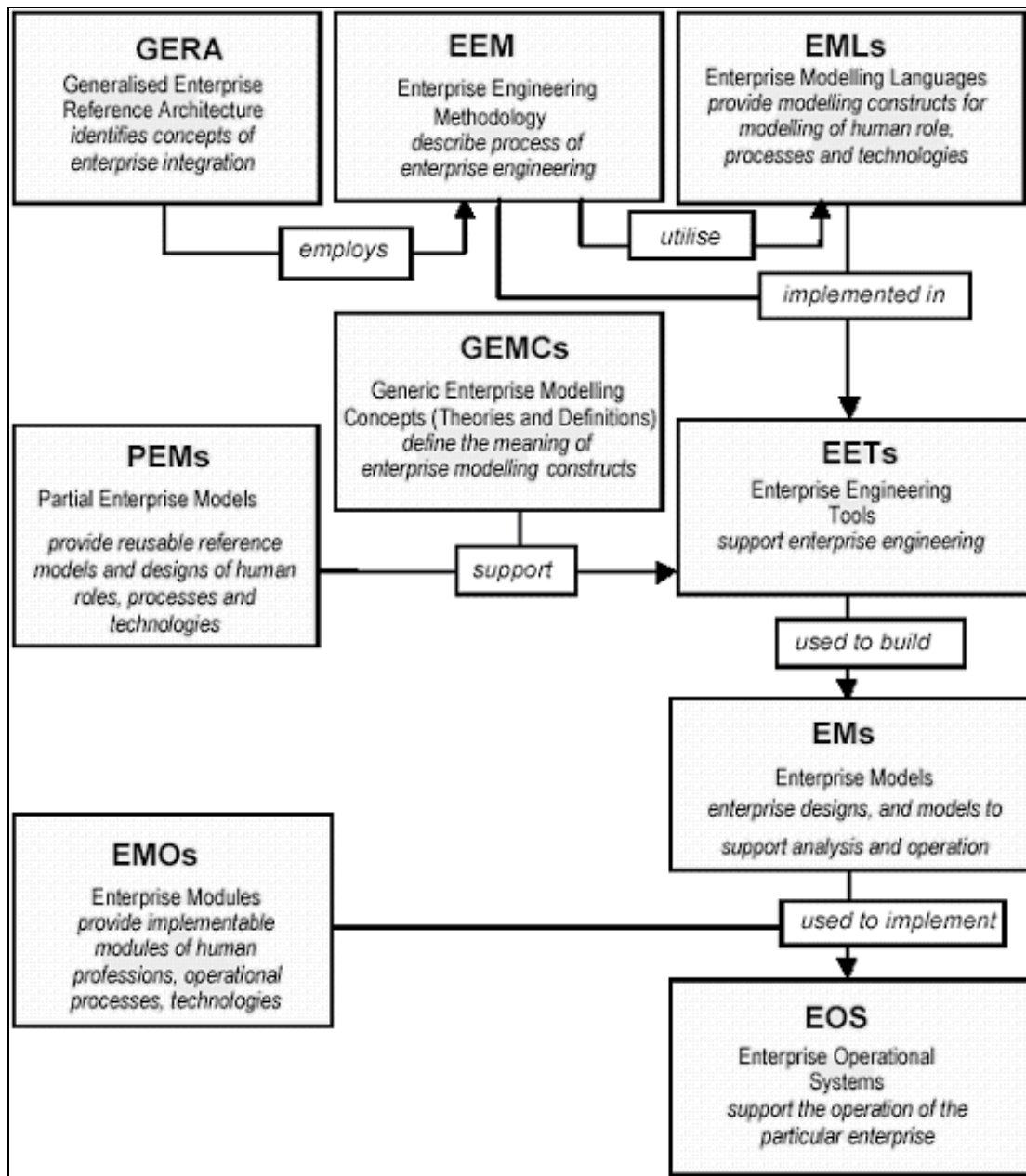


Figure 1. GERAM Framework Components (Source: GERAM V 1.6.3)

1.2. A Short Overview of TOGAF

TOGAF is an industry standard architecture framework that can be freely used by any enterprise developing enterprise architecture for use within [17]. While TOGAF version 7.0 focuses on the technical architecture, TOGAF Version 8.1 Enterprise Edition is applicable to all aspects of enterprise architecture development, namely, business process, information, application architecture, as well as technical architecture. Over the years, the United States Department of Defense (DOD) has spearheaded the development of EA as a discipline and as expected DOD developed the most of the early EA frameworks. The Technical Architecture

Framework for Information Management (TAFIM) was one the earliest formal EA framework. TAFIM essentially provides policies and guidelines to procure, develop and deploy information technology across the DOD. However due to certain obvious shortcomings in the TAFIM [13], the Command & Control, Communication and Computers Intelligence, Surveillance and Reconnaissance (C4ISR) model has overtaken TAFIM as the most widely accepted and practiced EA framework within the DOD. The current form of TOGAF has evolved from and has heavily borrowed some TAFIM concepts. TOGAF, also like many other EA frameworks provides several components that are depicted in Figure 2.

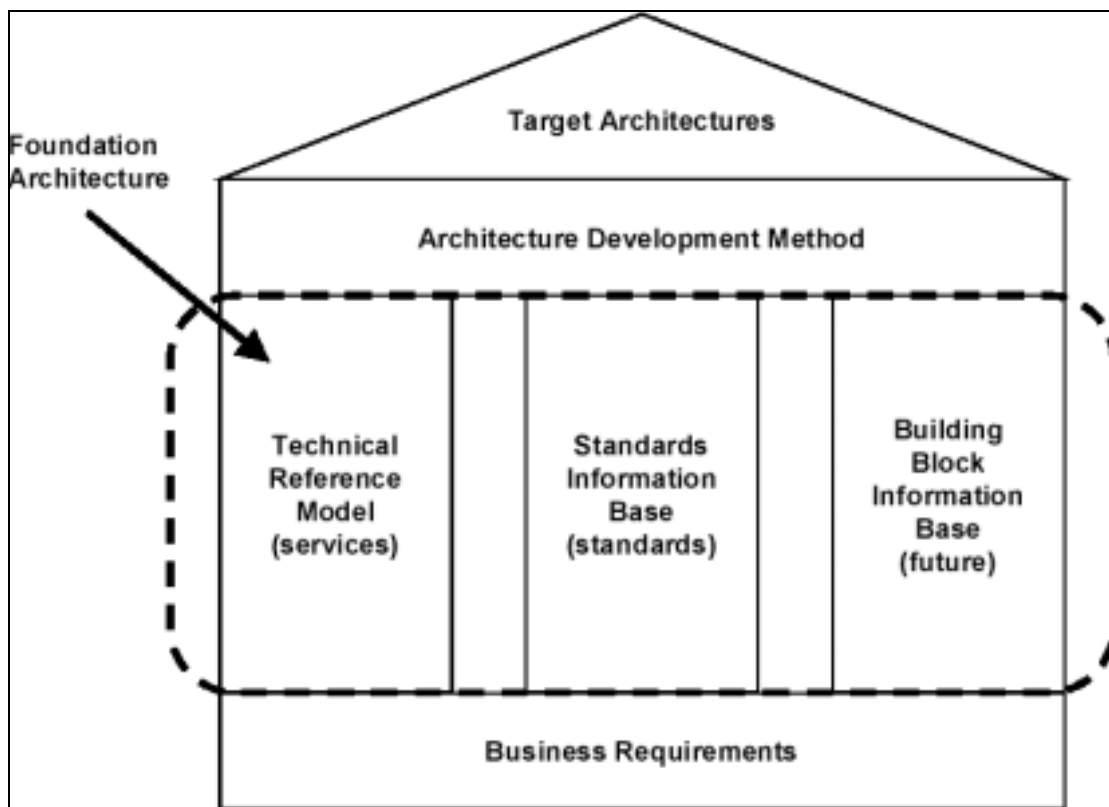


Figure 2. The TOGAF Components (Source: TOGAF V 8.1)

The scope of TOGAF V 8.1 Enterprise Edition encompasses all aspects of enterprise architecture development. This includes business process, information, technology and application architecture. The *business process architecture* specifies the organization's business strategies, goals and objectives and links these to business processes that are required to execute the identified strategies. The primary focus of the business process architecture is to create an integration mechanism between strategy formulation and strategy implementation. The *information*

*architecture*⁵ describes the structure of an organization's logical and physical data resources and management of such resources. The *technology architecture*⁶ describes the information technology infrastructure required to support the organization's business process and information architecture. The *application architecture*⁷ describes the software applications that are necessary to deploy organization's business processes governed by business rules [17].

2. Analysis Approach

As mentioned earlier, the main aim of this paper is to provide enterprise architects and engineers selecting TOGAF for implementation within their organization, with an understanding of GERAM requirements. This analysis would help in assessing TOGAF (as a candidate architecture) in meeting GERAM requirements and to check whether it is adequately equipped with all necessary components. In case of TOGAF lacking in certain areas, this paper provides guidance towards ensuring common understanding of deliverables, and ways to complete it using GERAM requirements as the reference. With this aim, the analysis centers on several aspects, each discussed in detail in Sections 3 to 9. Though there have been several attempts to map existing lifecycle architectures against one another [19], the difficulties in such a mapping makes analysis a challenging exercise. Subsequently, attempts to analyze architectures against fixed reference requirements improved the mapping outcomes, which resulted in a matrix like structure of requirements [4]. These requirements formed a critical input to GERAM document.

3. Life Cycle Phases

Typically, enterprise engineering literature points to existence of two main types of perspectives in architectural framework specification: the *static* perspective that represents the structural aspects at a specified point in time and the *dynamic* perspective that captures the various phases that architecture development goes through [12]. In the context of GERAM, the architecture life cycle phase⁸ usually means the activities and tasks performed during the architecture development process, the elements that constitute the inputs and outputs of each activity and the artifacts (or deliverables) that are produced as part of phase completion. Though phases could be visually represented in a sequential manner, in reality an

⁵ Also known as Data Architecture.

⁶ Specific focus of earlier versions of TOGAF, e.g. TOGAF 7.0.

⁷ Also known as Software Architecture.

⁸ This is specifically called the Generalized Enterprise Reference Architecture (GERA).

organization may operate in an iterative manner. However, the crucial point to be noted here is that life cycle phases in GERA do not represent the temporal aspect (i.e. succession in time), rather it is isolated from the 'time' aspect and represents only logical groupings of activities performed and artifacts produced.

3.1. TOGAF Life Cycle Aspect

TOGAF has a very well defined and explicit approach to life cycle aspect. This is called the TOGAF Architecture Development Method (ADM). With ADM, TOGAF takes a life cycle approach to architecture development [17]. ADM is a method for developing an EA to meet business and information technology needs of an enterprise. Given the iterative nature of architecture development process, the ADM is tightly integrated with TOGAF Enterprise Continuum (elaborated in Section 4). The Enterprise Continuum captures the temporal aspects of the architecture development that progressively guides the evolution of architectural artifacts through time as the EA initiative moves along the ADM phases. According to the TOGAF specification, each iteration of the ADM must include decisions regarding: (1) the breadth of coverage of the enterprise to be specified, (2) the depth of details to be specified, (3) the extent of time horizon aimed at and (4) the architectural assets to be leveraged in the organization's Enterprise Continuum. Figure 3 depicts the mapping between GERA lifecycle phases and TOGAF ADM phases.

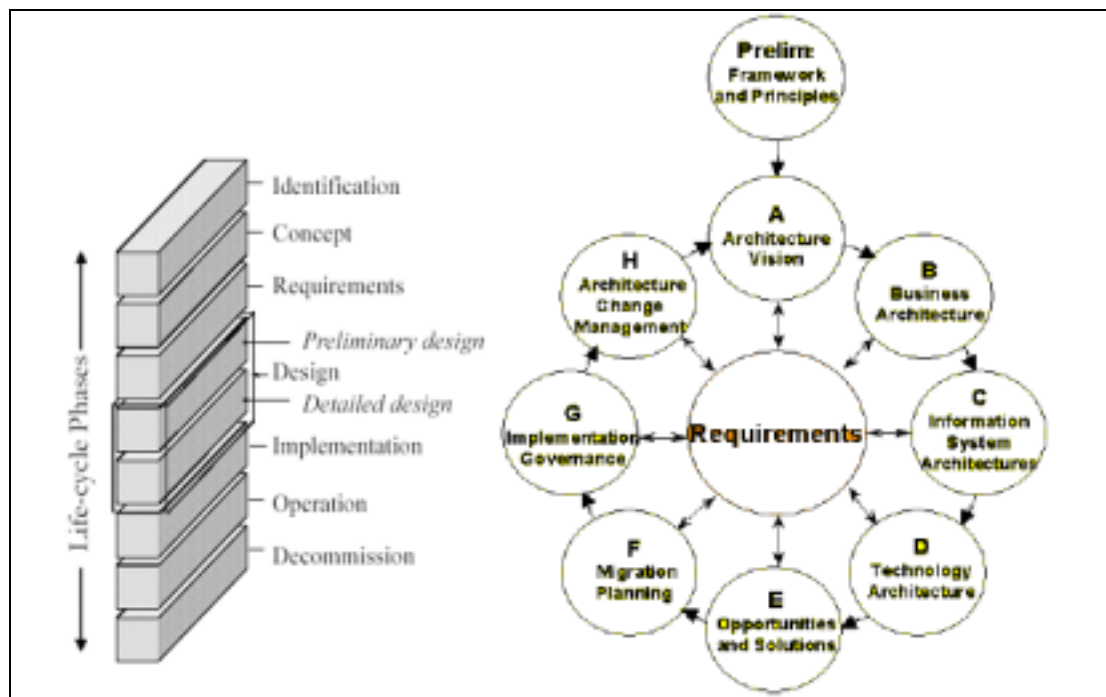


Figure 3. Mapping GERA Lifecycle and TOGAF ADM Phases

As is evident in Figure 3, the mapping of phases between GERA and ADM is largely direct. The *Identification* and *Concept* phases in GERA map to *Framework & Principles* and *Architecture Vision* in ADM. This is the initiation phase in the development of enterprise architecture where the organization (and the architect) attempt to understand the stakeholders' high level requirements and set the principles that would guide development in later phases. The *Requirements* phase in GERA maps to *Business Architecture* phase in ADM. In this phase the organization elaborates the business strategies, goals & objectives and specifies the business processes that are to be supported by the enterprise in order to fulfill business goals. The business case for architecture development is also created in this phase. The *Preliminary Design* phase in GERA maps to *Information Systems Architecture* and *Technology Architecture* phases of the ADM. This phase usually involves developing the information and the application architecture that would support the earlier described business architecture. Organizations can take data driven approach [15] or applications driven approach. Sometimes called the *Analysis* phase, the focus in this phase is more on the functional aspects of design (i.e. workflows and business rules governing them). The *Detailed Design* phase in GERA maps to *Technology Architecture* phase in ADM. The objective of this phase is to develop a detailed technical architecture that forms the basis for further implementation. This includes documenting the baseline technical architecture and the target technical architecture with a clear understanding of the gaps between the two. In this phase organizations usually take inputs from industry specific standards⁹ that may be available. The *Implementation* and *Operation* phases in GERA map to *Opportunities & Solutions*, *Migration Planning* and *Implementation Governance* phases in ADM. The focus in this phase shifts from detailed design specification to planning, implementation and continuous governance (operational support including maintenance) of the architecture in operation. While *Decommissioning* forms the last phase in GERA, ADM specifies *Architectural Change Management* as its final phase. The objective of this phase is to establish a process to create a new enterprise architecture baseline that is achieved with the completion of the *Implementation Governance* phase. This provides a framework to continually monitor the efficacy of the implemented architecture and initiate a new architecture development cycle when the organization feels the need.

⁹ For instance: Electronic Telecom Operations Model (eTOM) for the Telecommunications Industry, Health Level (HL7) for the Healthcare Industry and Petrochemical Open Software Corporation (POSC) for the Petrochemical Industry.

3.2. Concluding Remarks on Life Cycle Phases

The common understanding that any entity (including an enterprise) needs to be conceived, designed, implemented, operated and possibly retired (or revamped) is evident both in GERA and TOGAF. TOGAF through its ADM explicitly covers the life cycle aspect as required by GERAM and described in GERA. TOGAF specification provides elaborate descriptions of objectives, intent, approach, activities, artifacts, inputs and outputs for each phase.

4. Temporality and Succession

Besides life cycle phases any enterprise architecture development initiative progresses in time, hence the timeline related needs form a crucial part of the GERAM requirements. The critical difference between life cycle phases and timeline aspect is that the evolution of the enterprise architecture through time is not covered in the life cycle phases, which only indicates the logical groupings of activities and tasks performed. The temporal aspect allows an organization to evolve and adapt to changing business environment.

4.1. TOGAF Temporal Aspect

TOGAF explicitly acknowledges the existence of succession of steps during the life of the modeled entity that captures its progression, evolution and adaptation in time. This is done through a concept called the Enterprise Continuum. The Enterprise Continuum is tightly integrated with TOGAF ADM. While ADM specifies the process / methodology of moving from the foundation architecture to the enterprise specific architecture, the Enterprise Continuum provides time based anchors during the enterprise engineering lifecycle to depict various forms that the modeled entity takes. TOGAF categorizes the Enterprise Continuum into Architecture Continuum and Solutions Continuum. Figure 4 shows TOGAF Enterprise Continuum that also depicts the relationships between the Architecture and the Solutions Continuum.

While the Architecture Continuum focuses on progressive development of architectural building blocks, the Solutions Continuum represents the implementations of the architectures. The *Foundation Architecture* is the architecture of building blocks and standards that supports the complete information technology environment. It consists of general computing environment, general building blocks, technology standards for implementing the building blocks, overall direction for products and services and computing strategies.

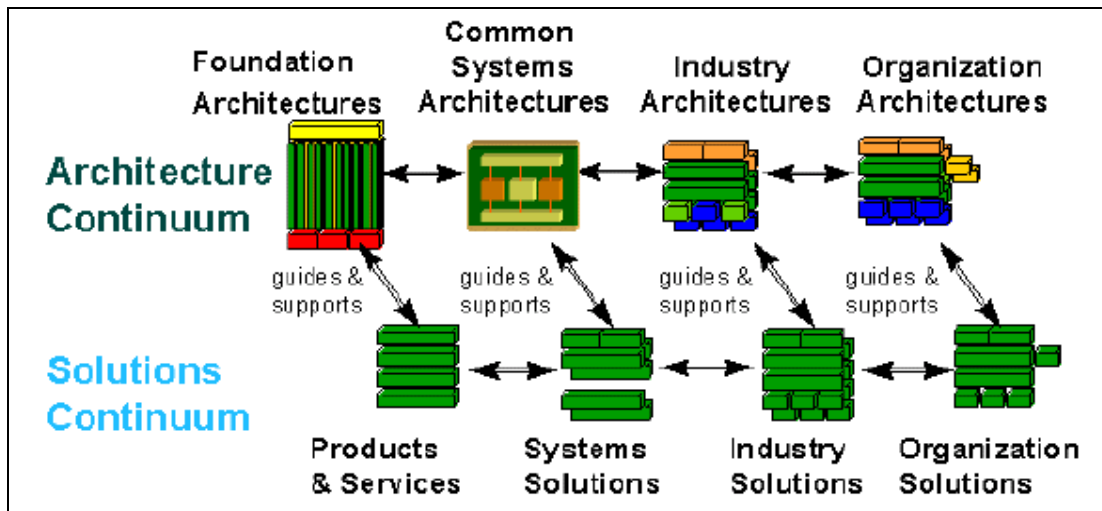


Figure 4. TOGAF Architecture Continuum (Source: TOGAF V 8.1)

Products that represent procurable hardware and software and *Services* representing training and consulting that are necessary to take advantage of the solutions support the *Foundation Architecture*. The *Common Systems Architecture* is the collection of specific services from the *Foundation Architecture* to assemble common solutions that are useful across several domains. Examples are Network Architecture, Security Architecture, and Management Architecture etc. These facilitate reuse of architecture across domains as they address generic requirements. *Systems Solutions* are an implementation of *Common Systems Architecture*, usually representing a common set of requirements and capabilities (horizontal requirements) that are not specific to any industry or organization. Examples include Customer Relationship Management (CRM) and Security System Products. The *Industry Specific Architecture* guides the integration of common system components to industry specific components. It reflects the requirements of a specific industry / vertical including industry specific business rules. *Industry Solutions* implement the *industry specific architecture* where the requirements tend to be vertical and not horizontal, though common to most organizations belonging to a specific industry. *Enterprise Architecture* is the organization specific architecture that is derived from all the earlier steps in the continuum. The EA is the most organization specific entity. *Enterprise Solution* is the highest level of organization specific implementation of the EA.

4.2. Concluding Remarks on Temporality & Succession

The concept of time-based progression is crucial to GERAM, and TOGAF explicitly covers this through its Enterprise Continuum. An entity that is modeled moves from generic to specific on the continuum allowing bidirectional traceability to

be maintained. As is seen in Figure 4, within the Enterprise Continuum, the relationship between the Architecture Continuum and the Solutions Continuum is one of guidance, support and direction [17], thereby synchronizing the progressive development to higher specificity along both dimensions. The Enterprise Continuum conceptualizes mechanisms to enhance productivity through leverage. The Architecture Continuum provides a consistent approach to understand different architectures and their components, while the Solutions Continuum views the different products, services and solutions needed to realize the architectures. Finally, it is crucial to understand that the start and finish of the Enterprise Continuum lie at the foundation architecture that acts as a toolset or repository of reusable guidelines and standards, and in this case it is the TOGAF Technical Reference Model and Standards Information Base [13].

5. Modeling Framework

The GERA Modeling Framework is a core component in the GERA Reference Architecture. The criticality of a modeling framework stems from the fact that it specifies the scope and type of models that need to be developed and maintained as part of the architecture development process. Analyzing TOGAF against modeling framework requirements would address two issues, which are:

- The extent of guidance provided in TOGAF with regard to the scope and list of models that need to be created and maintained
- The extent to which TOGAF is capable of accommodating the complete scope of modeling framework as per GERA requirements

5.1. TOGAF Modeling Framework

GERA modeling framework anchors around three dimensions: (1) the *lifecycle* dimension, (2) the *temporality and succession* dimension and (3) the *view* dimension. TOGAF directly addresses the first two dimensions through its Architecture Development Method that specifies a controlled modeling process of enterprise entities according to the life cycle activities and Enterprise Continuum that captures progressive specificity in creation of the EA along the time dimension, respectively.

With regard to the view dimension, TOGAF acknowledges the existence of multiple views or perspectives that are necessary to create a coherent specification of the EA, which is in line with GERA requirements. TOGAF recommends views for each of the four architectural categories. It explicitly defines the perspective from which a view is to be taken, the process of creating a view and recommendations to use a view. TOGAF recommended views and viewpoints include:

- *Business Architecture Views* that address the users of the system describes the business information flow and underlying business processes and business rules. Includes *People View, Business Process View, Business Function View, Business Information View, Usability View and Business Performance View*.
- *Information Architecture Views* that address the requirements of Database Designers and Database Administrators tasked with development, integration and maintenance of Information Resources. Includes *Data Flow View*.
- *Application Architecture Views* that address the needs of application / software development group (i.e. Systems and Software Engineers) tasked with development and maintenance of applications and components. Includes *Software Engineering View and Systems Engineering View*.
- *Technology Architecture Views* that address the requirements of personnel who are responsible for acquiring hardware and software infrastructure necessary to support all other groups. Includes *Communication Engineering View, Security View and Management View*.

Similarly GERA, with the intention of reducing model complexity splits EA into multiple views that represent the viewpoints of different stakeholders. In order to be GERAM compliant candidate architecture, GERA identifies the following model views (shown in Figures 5 and 6):

- *Entity Model Content Views*: Function, Information, Resource, Organization
- *Entity Purpose Views*: Customer Service & Product, Management & Control
- *Entity Implementation Views*: Human Performed Tasks, Automated Tasks
- *Entity Physical Manifestation Views*: Software, Hardware

While GERA does not necessitate every view to be present, there is a fairly direct mapping between GERA concept of views and TOGAF views. GERA *Entity Model Content Views* with focus on user oriented process representation of enterprise entity descriptions maps to TOGAF *Business Architecture Views*. GERA *Entity Physical Manifestation Views* with focus on software and hardware that are needed to realize the enterprise architecture maps to both TOGAF *Application* and *Technology Architecture Views*. GERA *Entity Implementation Views* addressing the identification of tasks that are / are not to be automated roughly maps to TOGAF *Application Architecture Views* where the workflows needed to support task automated could be specified. Lastly, GERA *Entity Purpose Views* with focus on mission of the enterprise entity and products & services needed to support enterprise objectives roughly map to TOGAF *Business Architecture Views*.

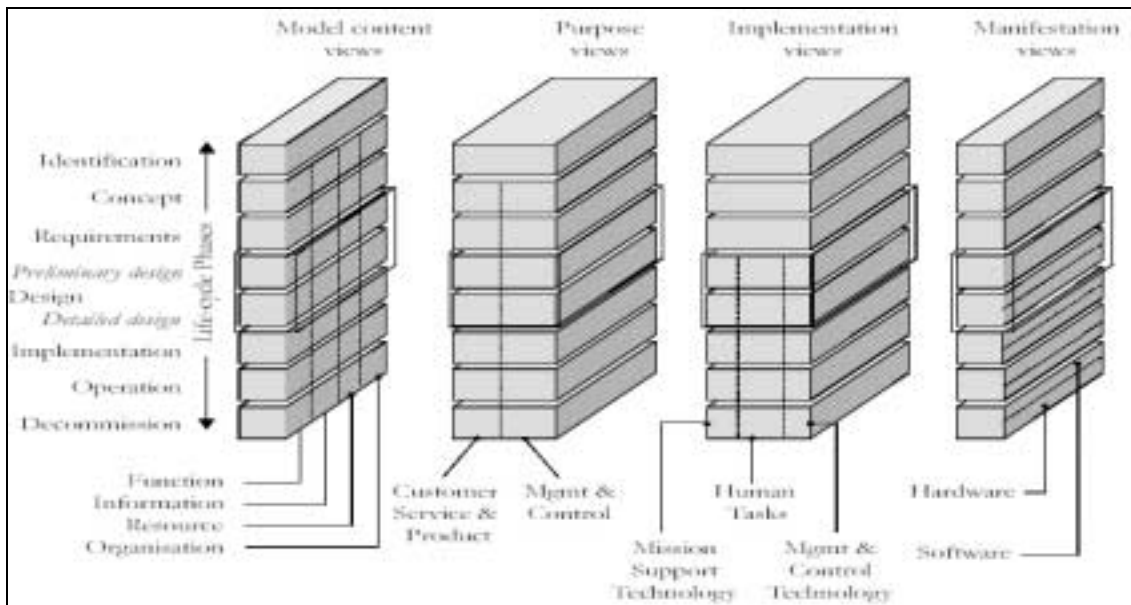


Figure 5. GERA Recommended View Types and Their Contents (Source: GERAM V 1.6.3)

5.2. Concluding Remarks on Modeling Framework

A modeling framework is a structure specifying the artifacts (deliverables) required to accomplish the modeling process. From the modeling framework point of view, in line with GERA requirements, TOGAF has a direct mapping vis-à-vis GERA lifecycle and temporality dimensions (as discussed earlier in Sections 3.1 and 4.1 respectively).

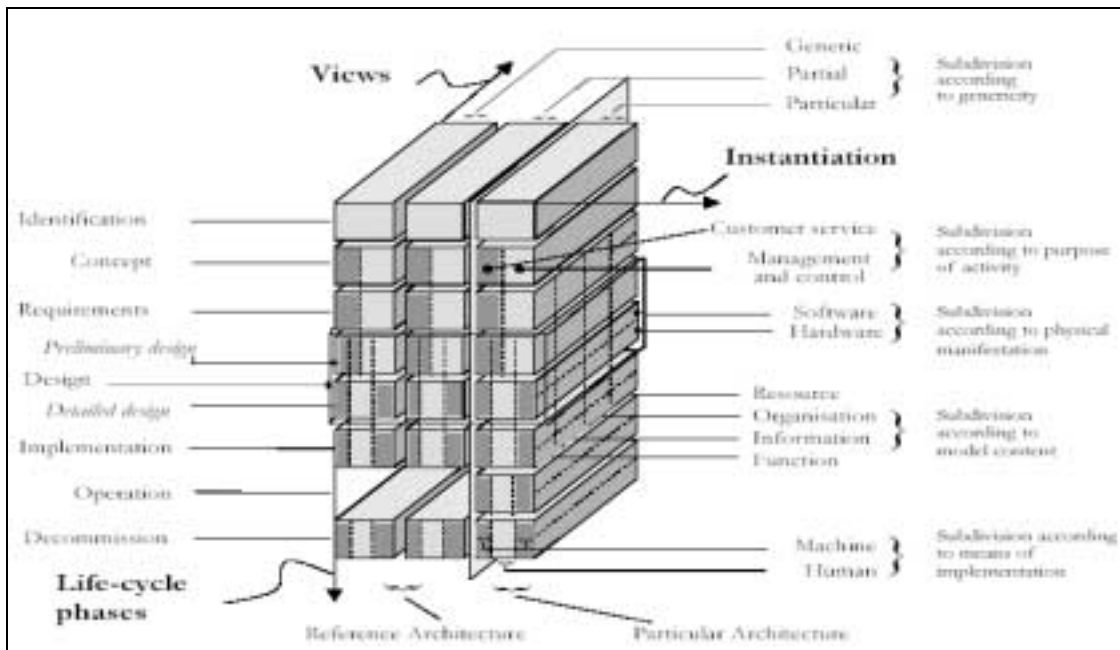


Figure 6. GERA Modeling Framework & Modeling Views (Source: GERAM V 1.6.3)

On the view dimension, while there are areas where the mapping is not direct, the point to be noted is the TOGAF taxonomy of recommended view does cover the entire intent of GERA view concepts. This can be interpreted as TOGAF fulfilling the GERA requirements as both explicitly mention that organizations are free to modifying existing or specify new views to meet specific requirements [9, 17]. TOGAF specifications do not limit the modeling framework to just models, but also includes other construct types like the Technical Reference Model (TRM), Standards Information Base (SIB), Architecture Building Blocks (ABB), Vocabulary, Integrated Information Infrastructure Reference Model (IIRM) and the Resource Base.

It is to be noted that an organization can take full benefit of an explicit modeling framework only if it is tightly integrated to the modeling methodology. In case of TOGAF, its ADM explicitly specifies the role of the modeling framework to support the enterprise architecture endeavor. While TOGAF provides recommendations in this regard, it is not prescriptive thus allowing breadth to the architect to configure the methodology and the framework as per specific requirements and needs.

6. Modeling Languages

The practice of enterprise architecture produces a set of artifacts (includes models, documents and other deliverables) aimed at specific audience with the intention of acting as a communication vehicle. In order to accomplish this communication enterprise models must be developed using languages that: (1) capture the intent of the aspect being modeled and (2) can be understood by the intended audience. A modeling language usually is an embodiment of modeling constructs, semantic and syntactic rules. Within the realms of information systems several formal modeling languages¹⁰ coexist for specific purposes. A modeling language must be able to maintain balance between complexity and expressiveness.

GERAM recognizes the fact that any enterprise architecture initiative is a large and complex undertaking and thus would require potentially multiple modeling languages [9]. Requirements specified in the GERAM are:

- The (set of) modeling language(s) must be able to represent every modeling view / viewpoint (as identified in Figure 6) for every enterprise architecture artifact in the development methodology and extent of specificity.
- Models developed for one view must be linkable with models in other views, if the linkage is a logical requirement for a coherent enterprise architecture view.

¹⁰ Includes: Unified Modeling Language (UML), Entity Relationship (ER) Diagram, Petri Nets, Event Driven Process Chains (EPC), Business Process Modeling Notation (BPMN), and IDEF Family of Languages etc.

- Modeling languages must be based on reliable ontologies (metamodels with semantic rules)

6.1. Modeling Languages for TOGAF

TOGAF, while specifying explicitly what artifacts need to be developed and when & how they are to be developed, does not define any dedicated architecture development language, nor does it try to recommend / prescribe any third-party modeling languages that might be appropriate. Enterprise architects must utilize several proprietary and third party modeling languages in order to accomplish all the architectural activities and artifacts. However, TOGAF does provide some guidance covering architectural best practices through its recommended architectural principles, patterns and governance strategies. Lack of the single / unified set of modeling language(s) creates the risk of low consistency and interoperability between the various models, and it is upon the architect to impose discipline in this regard. Table 1 provides some recommendations on potential languages that can be used to specify TOGAF artifacts¹¹.

6.2. Concluding Remarks on Modeling Languages

Given the heterogeneity of objectives underlying enterprise architecture endeavors, architects must choose modeling languages with which they are able to accomplish intended architectural activities. TOGAF does not recommend / specify any language for this purpose. Currently, there is no single modeling language capable of modeling all aspects of an enterprise. Hence, meaningful and complete enterprise model development needs several languages [18, 3]. Several general-purpose languages currently available are candidates and they include, Unified Modeling Language (UML)¹² for Application Architecture, Entity-Relationship (ER) Data Model using IDEF 1x for Information Architecture, Object – Role Model (ORM) and Object Constraint Language (OCL) for specifying business rules, Event Driven Process Chain (EPC) and Business Process Modeling Notation (BPMN) for specifying Business Process View, among others.

¹¹ Usually tool vendors support multiple modeling languages and diagrams to accomplish EA related activities. For instance Popkin's *System Architect v 9.1* provides many diagrams in its recommendations to be used. Available at (http://www.popkin.com/customers/customer_service_center/downloads/manuals/UserGuide.pdf).

¹² UML 2.0 supports 13 diagrams that can be used at various stages in the architecture development process, though these are most useful in specifying application / software architecture.

TOGAF Architectural Activities	Partial List of TOGAF Recommended Artifacts	Proposed Modeling Languages
Architecture Visioning	<ul style="list-style-type: none"> Request for architecture work Initial statement of architecture work Architecture principles Enterprise continuum Architecture vision Baseline architectures 	<ul style="list-style-type: none"> Rich pictures / English
Business Architecture	<ul style="list-style-type: none"> Statement of architecture work Business principles Target business architecture Business architecture views Gap analysis report 	<ul style="list-style-type: none"> Rich pictures / English UML System Dynamics BPMN / BPML OCL Structured English IDEF 0 & IDEF 3 ORM EPC
Information Architecture	<ul style="list-style-type: none"> Target information architecture Information architecture views Gap analysis report Impact analysis report 	<ul style="list-style-type: none"> IDEF 1 UML ERM ORM
Application Architecture	<ul style="list-style-type: none"> Target application architecture Application architecture views Gap analysis report Impact analysis report 	<ul style="list-style-type: none"> UML Structured English
Technology Architecture	<ul style="list-style-type: none"> Technology baseline description Technology principles Target technology architecture Gap analysis report Impact analysis report Technology architecture views 	<ul style="list-style-type: none"> TOGAF Format Rich pictures / English

Table 1. Possible Modeling Languages for TOGAF

However, using several languages not belonging to the same family (and not having the same underlying metamodel) creates challenges in maintaining consistency and interoperability. One potential solution currently under development is the proposed Unified Enterprise Modeling Language (UEML), which would be based on a common integrated set of metamodels, semantic rules and language constructs [18]. This initiative is still in its infancy and the hope to find ‘a language’ that is able to accomplish all of EA modeling objectives is farfetched. The architect must therefore choose language(s) appropriate as per project requirements.

7. Methodologies

Methodologies for enterprise engineering describe the processes of enterprise architecture. Typically, they are part of the lifecycle concept (covered earlier in

Section 3). A generalized methodology must be applicable to enterprises belonging to any industry. A methodology in the GERA sense, allows an enterprise to treat the EA endeavor as a project with specific deliverables. As seen earlier a formal methodology must address issues regarding: (1) the steps involved in the process, (2) scope and intent of each step, (3) input requirements at each step, (4) activities to be done as part of each step, (5) deliverables and guidance to develop them and (6) supporting materials that may be useful. GERA has specific requirements in each of these issues for any methodology to be a candidate enterprise engineering methodology. Additionally, GERA has several requirements for modeling methodologies, which are:

- The need to address human involvement and role in an EA initiative as it plays a very crucial part in the overall success of the program
- The necessity to differentiate between user-oriented and technology-oriented design, as this drives the extent of automation that needs to be supported by the enterprise
- The need to utilize and imbibe accepted project management techniques and practices as it increases the extent of management and control that can be exercised
- The need to take economic and performance factors into consideration as this helps justify the initiative and aligns it with business objectives

7.1. TOGAF Methodology

As described in Section 3.1, TOGAF ADM is a comprehensive modeling methodology that meets all requirements of a detailed step-by-step approach as mandated by GERA. Besides providing detailed architecture development process specification in its ADM, TOGAF provides comprehensive guidance and inputs that meets GERA additional requirements for methodologies.

TOGAF recommends use of *Business Scenarios* in order to elicit high-level requirements in the *Architecture Vision* and *Business Architecture* phases. It also provides a recommended process to develop such scenarios that aim to capture the business process and applications enabled by the architecture, the business and technology environment, the human and computing entities and the desired outcome of proper execution. The critical point to be noted here is that TOGAF *Business Scenarios* explicitly place significance to human 'actors' vis-à-vis computer actors and differentiation between business (i.e. user oriented) and technology-oriented design, thus completely fulfilling GERA Methodology requirements.

The establishment and operation of an *Architecture Board* is deemed crucial for overall program success. This board, besides providing management oversight to the complete EA initiative aims to create consistency between sub architectures, identify reusable components, ensure flexibility of the enterprise's architecture, enforce architectural compliance, enhance organization's architectural maturity, ensure discipline in adoption of ADM, ensure fulfillment of all contractual obligations, keep organization's IT objectives aligned with business objectives, validate reported service levels and cost savings, monitor the EA's business case commitments and provide advice, guidance and information [17]. TOGAF provides architecture review checklist for hardware and operating systems, software services and middleware, applications, information management, security, system management, overall architecture and methods and tools for the Architecture Board to use to accomplish its objectives.

In order to ensure that there is no ambiguity between expectations of the program sponsors and the architecture development team, TOGAF recommends the use of *Architecture Contracts* that aim to build management practice of continuous monitoring of integrity, changes, decision making and audit of all architecture related activities, adherence to principles, standards and requirements, management of risks and ensure accountability and discipline. Typically TOGAF recommends that *Architecture Contracts* minimally contain the overall nature of the agreement, scope of the architecture, principles and requirements, conformance requirements, architecture development process and phases, target architecture measures, deliverables and architecture delivery and business metrics. Existence of an Architecture Board and an Architecture Contract implicitly means that TOGAF expects organizations to follow a project based approach to EA development and economic and performance factors¹³ are duly taken into consideration as required by GERA. This is further made clear with Phase F (i.e. Migration Planning) of the ADM which mandates formal project initiation, cost-benefit analysis, project prioritization, project roadmap and project execution. TOGAF migration planning phase is depicted in Figure 7.

With the aim of ensuring that enterprise architecture is managed and controlled at an enterprise-wide level, TOGAF recommends having a proper *Architecture Governance* framework in place. It further requires organizations to specify

¹³ Organizations can develop IT performance scorecard modeled along the Balanced Scorecard approach. The IT scorecard usually looks at IT performance from four perspectives: *Corporate Contribution*, *Customer Orientation*, *Operational Excellence* and *Future Orientation* [16].

Architecture Maturity Models, Critical Success Factors, Key Goal Indicators and Key Performance Indicators as part of the governance framework [17].

7.2. Concluding Remarks on Methodologies

With guidance provided by ADM, organizations need to build in house capability to conceptualize, design, develop, deploy and maintain the enterprise architecture. In order to realize these goals, GERA mandates specific requirements that EA frameworks must fulfill. TOGAF *Business Scenarios*, *Architecture Board*, *Architecture Contracts* and *Architecture Governance Framework* are rich sources of input to enable organizations to make informed and reasoned decisions and enforce discipline. The five level *Architecture Maturity Model* further reinforces the efficacy of the methodology in improving organization's architectural practices.

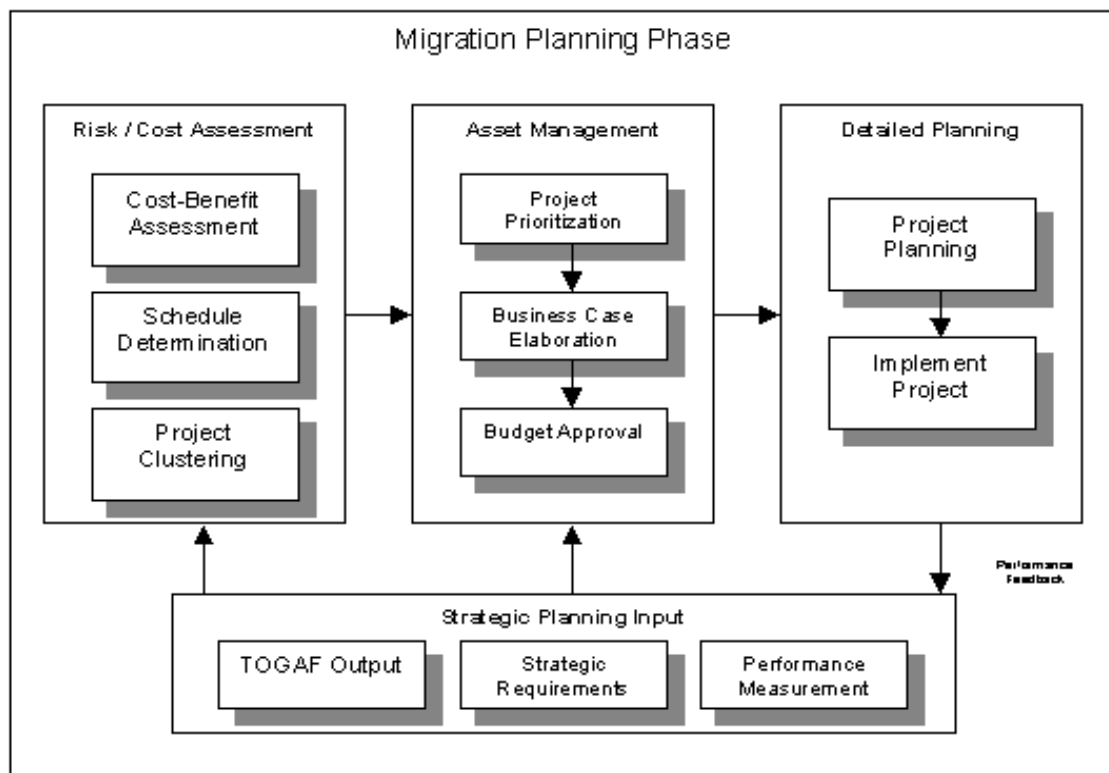


Figure 7. TOGAF Migration Planning Phase (Source: Perks & Beveridge, 2003)

8. Reference Models

Primary drivers for enterprise engineering include knowledge creation and its management [12]. Good enterprise architecture has the capability to capture, classify and encapsulate enterprise knowledge. Management of enterprise knowledge is a justifiable activity only if it leads to *complete* or *partial* reuse of that knowledge. In enterprise engineering, reuse is achieved through the use of Reference Models (also called Partial Models in GERAM parlance). Typically as the specificity of a reference

model increases, its applicability within the architecture develop process becomes narrower and more focused. The extent of reuse thus depends on two dimensions, i.e. the depth of reuse and the breadth of reuse. A generic reference model achieves broader reusability scope, whilst a specific reference model achieves deeper scope of reusability. Reference models are used to increase overall modeling efficiencies, however still requiring organization specific adaptation. Partial Models in the GERA sense may capture some common part of a class of enterprises, represent prototypes or abstract models that require specialization and adaptation before being realized as enterprise specific models. GERA specifies four types of partial models, which are: (1) partial human role models, (2) partial process models, (3) partial technology models and (4) partial models of IT systems.

8.1. TOGAF Reference Models

TOGAF provides several reference models as guidance to the enterprise architecture development process. As seen in Figure 2, the *Technical Reference Model*, the *Standards Information Base* and the *Building Block Information Base* form the three pillars of TOGAF foundation architecture.

TOGAF foundation architecture (the left most entity in Figure 4) is a collection of generic technology services (groups of functionalities) that organizations can adapt from for specific uses. It is possible that some organizations find the generic services completely describe their technical architecture. The process of evolving the foundation architecture into organization specific architecture is the Architecture Continuum (see Section 4.1). The process of controlling this evolution is the ADM. TOGAF TRM is simply a catalog of all technical functions (called services) needed to support business systems. It contains *taxonomy* with terminology and provides a coherent description of components and the conceptual structure of an information system and an associated TRM graphic, which is a visual representation of the taxonomy shown in Figure 8. In the GERA sense, the TRM maps to *Partial Technology Model* and *Partial Models of IT Systems*.

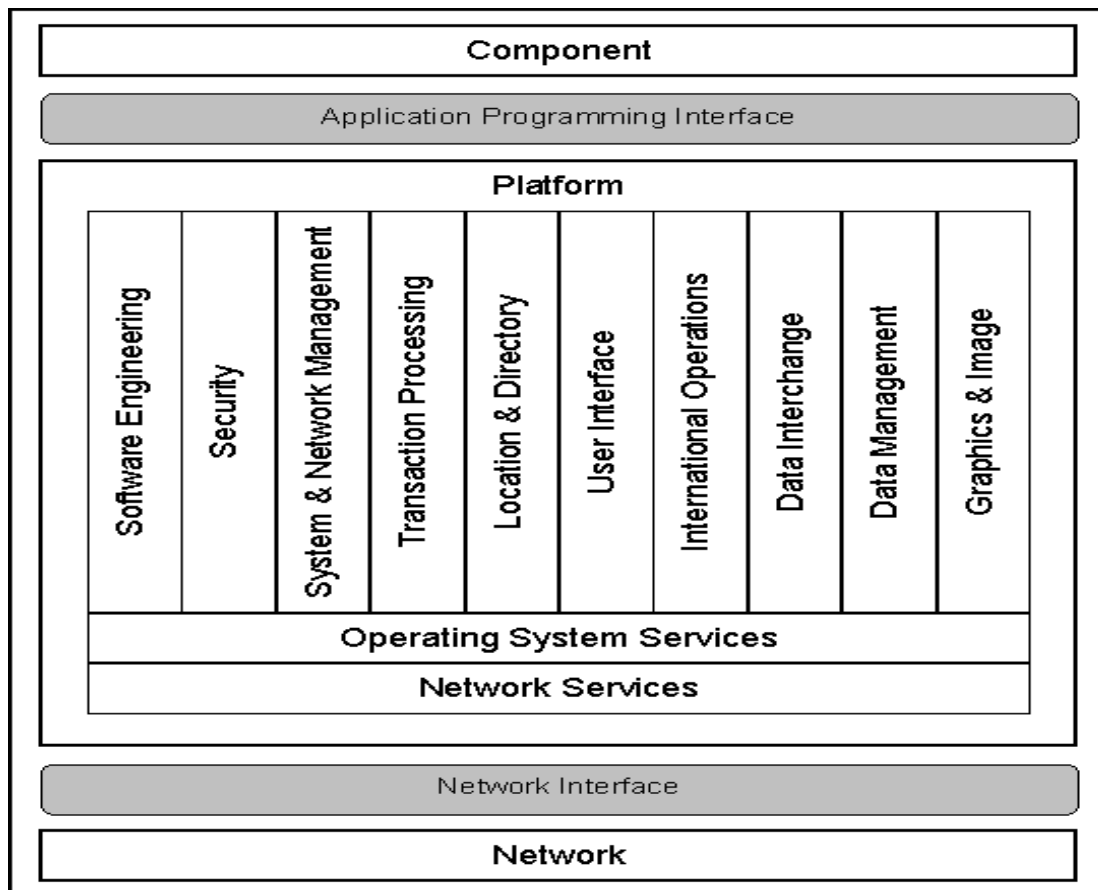


Figure 8. TOGAF Technical Reference Model (Source: TOGAF V 8.1)

The Standards Information Base is a catalog of technology standards and specifications that are useful in implementing the services identified in the TRM. The primary purpose of the SIB is to provide inputs to the architect during the development process to populate the architecture with technologies and products that meet TOGAF requirements. The SIB also provides guidance for procurement and acquisition of compliant products to realize the architecture. It primarily meets the GERA *Partial IT System Model* requirements.

The Architecture Building Block (ABB) Information Base is a collection of building blocks that aid in the assembly of the target architecture to meet business needs. A building block is simply a bundle of functionality specified to meet business needs, which provides published interfaces for other building blocks to access the functionalities. This makes the architecture a group of building blocks shown in an architectural model, with integration (assembly) specifications to meet business requirements. Hence the architecture will contain building blocks to implement only those services that are required conforming the desired standards. The primary goals of a building block based assembly approach to enterprise architecture are to

promote reusability, integration and interoperability [17]. This maps to GERA *Partial Technology Model* requirements.

While TOGAF recognizes the importance of business processes in the architecture development method through its guidance on development of *business scenarios*, it explicitly does not provide any reference models in this area. Thus an enterprise must scout for other sources for such information which may include for instance domain specific business process reference models like *Straight Through Processing (STP)* for the Financial Securities Industry, *Collaborative Planning, Forecasting and Replenishment (CPFR)* for the Retail Industry and *Supply Chain Operations Reference (SCOR) Model* for the Manufacturing Industry.

TOGAF addresses the GERA *Partial Human Role Model* requirements through its architecture viewpoints, which looks at enterprise models from a specific stakeholder's perspective. At the highest-level TOGAF identifies four stakeholders: (1) users, (2) systems and software engineers, (3) operators, administrators and managers and (4) acquirers. Enterprise architecture's multiple views relating to specific viewpoints provide insights into stakeholder responsibilities and authorizations.

8.2. Concluding Remarks on Reference Models

Enterprise architecture is a complex undertaking, and in their endeavor to tackle the complexity issue, architects (and enterprises) often use the 'divide and rule' policy by way of creating several views, each addressing a specific viewpoint, resulting in multiple models. Hence to be GERAM compliant, reference architectures like TOGAF must provide constructs, frameworks and guidelines to develop these models. Partial models play a significant role in reducing this complexity by providing 'templates' that can accelerate the architecture initiative. Partial models are created either by specializing generic models or generalizing specific models, thus requiring a strong validation process. Enterprise model verification and validation is a complex task necessitating a formal approach in most cases [5]. TOGAF in largely meets the GERA *Partial Model* requirements, though mechanism to create executable partial models would greatly enhance architectural standards [12].

9. Enterprise Engineering Tools

Given the complexity of EA initiatives, Computer Aided Enterprise Engineering (CAEE) tools are necessary to develop, operationalize and manage enterprise models. GERAM has several high level requirements for CAEE tools. These are:

- Support for analysis and evaluation of enterprise models allowing model enactment through simulation
- Comprehensive user guidance and embedded architecture development process
- Support for collaborative as well as individual design and engineering activities
- Provision of shared repository of all reference models, informal design descriptions and patterns, document etc.
- Ability to incorporate engineering of both products and enterprises
- Forward and reverse (complete round trip) engineering capabilities

9.1. Tools for Enterprise Engineering

Absence of a common / unified language for enterprise engineering and existence of several competing reference architectures¹⁴ (with no dominating player) has given rise to myriad of tools that claim to do ‘comprehensive’ enterprise engineering. Due to business reasons it is not viable for any tool vendor to just support one reference architecture like TOGAF. Hence tools routinely support most of the commonly popular reference architectures in a bid to improve tool marketability and acceptance. Table 2 provides a summary of tools and their generic enterprise engineering capabilities [1]. Note that ‘++’ depicts primary functionality and ‘+’ depicts secondary functionality.

9.2. Concluding Remarks on Modeling Tools

As is evident from Table 2, it is difficult to find a tool that provides all EA development capabilities and hence organizations use a set of tools to accomplish the task. The challenge with this approach is model interoperability and integration. All tools mentioned in Table 2 do not support TOGAF, but there are tools (like System Architect) with explicit claims for TOGAF support. A comprehensive review of commercial third party and proprietary tools is beyond the scope of this paper.

¹⁴ The major ones include: *Zachman Framework for Enterprise Architecture, Reference Model for Open Distributed Processing, Model Driven Architecture, C4ISR Architecture Framework, Purdue Enterprise Reference Architecture, GRAI Integrated Methodology, Computer Integrated Manufacturing Open Systems Architecture and Architecture for Integrated Information Systems.*

	PROCESS & ORGANIZATION MODELING	SOFTWARE & SYSTEM MODELING	INFORMATION MANAGEMENT MODELING	REPOSITORY	MANAGEMENT
ARIS TOOLSET	++				
ASG-ROCHADE				++	
ENTERPRISE ARCHITECT	+	++			
FRONT ARENA ENTERPRISE				+	++
IBM S		+	++		
METIS	++	+	+	+	
MS VISIO 2002 PROFESSIONAL		++	+		
MS VISUAL STUDIO .NET ENTERPRISE ARCHITECT		++			
PTECH FRAMEWORK	++	+	+	+	
IBM RATIONAL ROSE		++			
SELECT ENTERPRISE	+	++		+	
SYSTEM ARCHITECT	++	++			
TIVOLI ENTERPRISE					++

Table 2. List of Enterprise Engineering Tools and Their Capabilities

10. Other Aspects

This last section covers rest of the GERAM requirements, primarily entity types and their recursion and enterprise modules. As per GERAM, an essential activity in enterprise architecture is the identification of entity types and relationship between them. An entity is a logical organization, which could be a department, division or an entire organization, with a life history. This provides the means for entity classification required for a more structured approach to EA. GERA introduces the concept of five entity types, which are:

- Strategic enterprise management entity
- Enterprise engineering entity
- Enterprise entity
- Product entity
- Methodology entity

Figure 9 depicts the relationships between the lifecycles of GERA entity types.

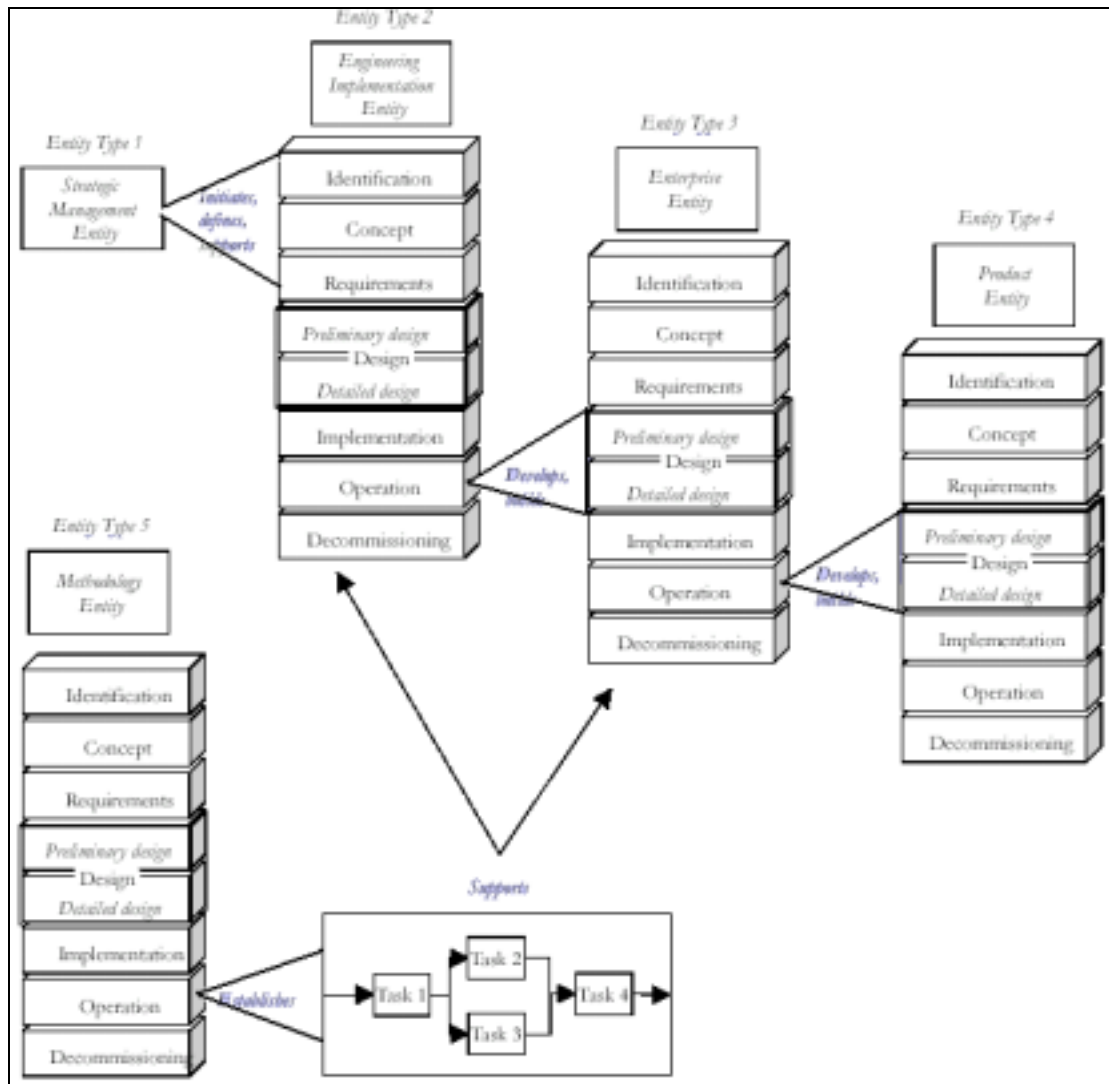


Figure 9. Relationships Between GERA Entity Types (Source: GERAM V 1.6.3)

GERA also identifies the concept of Recursivity of entity types, where Recursivity is defined as the direct and active influence of one entity in the development of another entity.

10.1. Entity Types and Their Recursivity in TOGAF

Though TOGAF does not explicitly mention entity types and their recursion, the framework is compatible to the concept because it does have a very strong concept in Enterprise Continuum (i.e. Architecture and Solutions Continuum), which directly supports the recursion concept. It is clearly evident that as the enterprise moves in time through the continuum, different groups may perform specific activities and there is bidirectional traceability along the continuum.

10.2. Enterprise Modules in TOGAF

Similar to the concept of component based development in software engineering, GERAM also recommends the use of pre built 'components' / 'modules' and incorporate an assembly based approach to enterprise architecture. This not only allows organizations to save time, but also build complex architectures and take advantage of engineering best practices. As seen in Figure 2, this is explicitly mentioned in TOGAF through its building blocks concept, though at present TOGAF has not yet specified any building blocks and its elaboration is conceptual (still to be put in practice). In fact TOGAF goes a step further and acknowledges the need for architectural patterns, which may be considered one abstraction level higher than building blocks. TOGAF considers patterns as a way to assemble building blocks within a specific problem context [17]. However, like building blocks, TOGAF does not provide any specific architectural patterns that can be put in practice, yet.

11. Final Conclusions

GERAM / ISO IS 15704: 2000 provides a generalized set of requirements against which several reference architectural specifications (like TOGAF) may be mapped. It is not to be viewed as a competitor to TOGAF or to any other reference architecture. Rather, it represents common baseline requirements that constitute enterprise architecture and engineering. Such a baseline allows enterprises and architects to assess various candidate reference architectures and choose the most appropriate one based on their specific business needs. This makes the rationale for choosing a specific architecture reasoned and rational and not impulsive and swayed by hype.

The analysis performed in this paper has shown that TOGAF meets GERAM requirements to a very large extent and areas where it is still 'weak' are largely limited to concepts that TOGAF recognizes but is yet to still completely specify. However two points to be noted are:

- TOGAF is still evolving and it is highly likely that eventually it would map to GERAM even further, as a common set of requirements also has the potential to not only assess but assist in the evolution of reference architectures [12]
- The discipline of enterprise architecture and engineering is also evolving, hence every reference architecture 'hopes' to set trends rather than follow it, in order to maintain its comparative advantage and edge.

Some of the recurring themes in EA include reducing complexity, increasing standardization, and perpetuating best practices. An effective EA places heavy

emphasis on standardization since it not only encourages component reuse but also brings down the total cost of ownership of a system. Standardization helps a firm to ultimately reap the benefits of economies of scale and scope. However, advancement of EA discipline to address these themes aided by open specifications like TOGAF is limited by the absence of a unified architectural development language. If history is anything to go by, then success of Object Oriented Analysis and Design is largely attributable to the Unified Process and the Unified Modeling Language. Going by the same yardstick, success and advancement of Enterprise Architecture and Engineering will be greatly aided if reference architectures like TOGAF and architectural development languages like UEML come together and provide the necessary impetus.

References

- [1] F. Arbab, M. Bonsangue, J.G. Scholten, M-E. Iacob, H. Jonkers, M. Lankhorst, E. Proper, A. Stam, State of the Art in Architecture Framework and Tools, Telematica Institut Version 1.2, 2002.
- [2] P. Bernus, Some thoughts on enterprise modeling, International Journal of Production Planning and Control 12 (2), 2001, pp. 110 –118.
- [3] P. Bernus, L. Nemes, G. Schmidt (Eds.), Handbook on Enterprise Architecture, Springer-Verlag, Berlin, Heidelberg, 2003.
- [4] P. Bernus, L. Nemes, T.J. Williams (Eds.), Architectures for Enterprise Integration, Chapman & Hall, London, 1996.
- [5] V. Chapurlat, B. Kamsu-Foguem, F. Brunet, Enterprise model verification and validation: an approach, Annual Reviews in Control 27, 2003, pp. 185 – 197.
- [6] D. Chen, G. Doumeingts, The GRAI-GIM Reference Model, Architecture and Methodology, In: P. Bernus, L. Nemes, T.J. Williams (Eds.), Architectures for Enterprise Integration, Chapman & Hall, London, 1996.
- [7] D. Chen, B. Vallespir, G. Doumeingts, GRAI Integrated Methodology and its mapping onto generic enterprise reference architecture and methodology, Computers in Industry 33 (3), 1997, pp. 387 – 394.
- [8] CIMOSA Association, CIMOSA – Open System Architecture for CIM Technical Baseline, Version 3.2, 1996.
- [9] IFIP-IFAC Task Force, GERAM: Generalized Enterprise Reference Architecture and Methodology, IFIP-IFAC Task Force on Architectures for Enterprise Integration March Version 1.6.3 (March), 1999.

- [10] A.H. Lewis, L.W. Wagenhals, C4ISR Architectures I: Developing a process for C4ISR Architecture Design, Journal of INCOSE 3 (4), 2000.
- [11] H. Li, T.J. Williams, A formalization and extension of the Purdue Enterprise Reference Architecture and the Purdue Methodology (Report 158), Purdue Laboratory for Applied Industrial Control, West Lafayette, Indiana, 1994.
- [12] O. Noran, An Analysis of the Zachman Framework for enterprise architecture from the GERAM perspective, Annual Reviews in Control 27, 2003, pp. 163 – 183.
- [13] C. Perks, T. Beveridge, Guide to Enterprise IT Architecture, Springer-Verlag, New York, NY, 2003.
- [14] A.W. Scheer, Architecture for Integrated Information Systems, Springer-Verlag, Berlin, Heidelberg, 1992.
- [15] S.H. Spewak, Enterprise Architecture Planning 2nd Edition, John Wiley & Sons, New York, NY, 2002.
- [16] The Gartner Group, The IT Scorecard Program Introduction and Overview, 1998.
- [17] The Open Group, The Open Group Architectural Framework Enterprise Edition, Version 8.1, 2003. Available (<http://www.opengroup.org/architecture/togaf/#download>)
- [18] F.B. Vernadat, The CIMOSA Languages, In: P. Bernus, K. Mertins, G. Schmidt (Eds.), Handbook of Information Systems, Springer-Verlag, Berlin, Heidelberg, 1998, pp. 243 – 263.
- [19] T.J. Williams, D. Zoetekouw, J. P. Shewchuck, D. Chen, H. Li, Techniques to map the architectures directly against one another, In: P. Bernus, L. Nemes, T.J. Williams (Eds.), Architectures for Enterprise Integration, Chapman & Hall, London, 1996.
- [20] J.A. Zachman, A Framework for information systems architecture, IBM Systems Journal 26 (3), 1987, pp. 276 – 292.
- [21] J.A. Zachman, Enterprise Architecture – A Framework, ZIFA, 2000. Available (<http://www.zifa.com>)