
**Enterprise modelling and
architecture — Requirements for
enterprise-referencing architectures
and methodologies**

*Modélisation et architecture d'entreprise — Exigences pour les
architectures et les méthodologies de référencement d'entreprise*

STANDARDSISO.COM : Click to view the full PDF of ISO 15704:2019



STANDARDSISO.COM : Click to view the full PDF of ISO 15704:2019



COPYRIGHT PROTECTED DOCUMENT

© ISO 2019

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

	Page
Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Abbreviated terms	5
5 Requirements for enterprise-referencing architectures and models	6
5.1 General requirements	6
5.2 Applicability and coverage of enterprise architecture	7
5.2.1 Kinds of enterprise	7
5.2.2 Characterizing enterprise architecture	7
5.2.3 Enterprise engineering and architecting methodology	7
5.2.4 Enterprise design	8
5.2.5 Enterprise operation	8
5.3 Essential concepts for enterprise-referencing architecture	8
5.3.1 Span of conceptual orientation	8
5.3.2 Human oriented	8
5.3.3 Process oriented	8
5.3.4 Interoperation oriented	8
5.3.5 Decision oriented	8
5.3.6 Realization oriented	9
5.3.7 Technology oriented	9
5.3.8 Environment oriented	9
5.3.9 Lifetime oriented	9
5.3.10 Stakeholder oriented	10
5.3.11 Viewpoint oriented	10
5.3.12 Model oriented	11
5.3.13 Model view oriented	12
5.3.14 Enterprise interoperation oriented	13
5.3.15 Verification and validation oriented	13
6 Components of enterprise-referencing architectures	13
6.1 Enterprise-referencing models	13
6.1.1 Purpose of enterprise-referencing models	13
6.1.2 Kinds of enterprise-referencing models	14
6.2 Modelling languages	16
6.2.1 Requirements for modelling languages and constructs	16
6.2.2 Expressiveness	16
6.2.3 Semantics and syntax of an enterprise-referencing model	16
6.2.4 Names, labels and glossary	17
6.2.5 Elements of interoperation	17
6.3 Models as representations	18
6.3.1 Representing enterprise characteristics	18
6.3.2 Concepts of internal structure	18
6.3.3 Compatibility of structuring approaches	18
6.3.4 Concepts of enterprise-referencing behaviour	19
6.3.5 Short-term and long-term behavioural change	19
6.3.6 Representation of behaviour	20
6.3.7 Concepts of hierarchy	20
6.3.8 Recursion in decomposition	21
6.3.9 Iteration	22
6.3.10 Availability and format of model information	22
6.3.11 Management of constituent parts	23

6.4	Impact of genericity	23
6.4.1	Generic enterprise elements	23
6.4.2	Partial enterprise models	23
6.4.3	Particular enterprise models	23
6.5	Enterprise perspectives and viewpoints	24
6.5.1	Primary concern perspectives	24
6.5.2	Additional concern perspectives	26
6.6	Enterprise-referencing modelling framework	26
6.7	Tools	27
6.8	Modules	27
6.9	Enterprise operational systems	27
6.10	Representation	27
Annex A	(informative) Key principles of enterprise integration and interoperation	28
Annex B	(informative) Generalized Enterprise Reference Architecture and Methodology (GERAM)	32
Annex C	(informative) Relationship of this document with other International Standards related to architecture for enterprise systems	67
Bibliography		70

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 5, *Interoperability, integration, and architecture for enterprise systems and automation applications*.

This second edition cancels and replaces the first edition (ISO 15704:2000), which has been technically revised. It also incorporates the Amendment ISO 15704:2000/Amd.1:2005.

The main changes compared to the previous edition are as follows:

- alignment of terminology for consistency with other standards developed by ISO/TC 184/SC 5;
- incorporation of the rules and guidelines for modelling from ISO 14258 to express enterprise-referencing architecture methodologies;
- revision of content related to recursive structure and iterative methods;
- redrafting of life history concept to address enterprise change management;
- discussion of modelling dimensions necessary for user views, including the extent of detail and composite models that span dimensions;
- harmonization with ISO/IEC/IEEE 42010 and other International Standards related to architecture for enterprise systems;
- inclusion of discussion concerning the relationship of this document with other International Standards related to architecture for enterprise systems (ISO/IEC 10746, IEC 62264, ISO 15745);
- updating of Annex B on Generalized Enterprise Reference Architecture and Methodology (GERAM) in accordance with GERAM 1.6.3.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

0.1 Rationale for enterprise architectures and models

Industrial and information age enterprises create and modify manufacturing and business operations to improve performance in local and global markets. In operation they deploy a variety of resources including people, information systems, automated machinery and business services. Individually and collectively these resources provide the functional capabilities required to perform manufacturing and business processes and their constituent activities. The arrangement, targeting and interworking of resources need to accomplish the enterprise mission, which requires suitable business rules and organizational structures that enable the enterprise to provide products and services in conformance with agreed upon criteria.

Enterprises operate under uncertain and changing market and environmental conditions that make ongoing enterprise engineering beneficial. Enterprises cooperate within a heterogeneous environment with multiple constituents using different models and systems. In addition, most enterprises operate within the context of another enterprise with which it interoperates, as with a supply chain or the parent company. It follows that enterprise personnel have a variety of responsibilities in the conception and ongoing development of the mission, business rules, manufacturing and business processes, organizational structures, and supporting resources and services. Because of the complexity involved in enterprise engineering, invariably it becomes necessary to deploy means of assessing, structuring, coordinating and supporting these engineering activities, including means for collaboration support and interoperation.

Generalized enterprise-referencing architecture concerns and components of enterprise modelling constitute a reference base that provides a generally applicable means of arranging and coordinating enterprise engineering and associated technology development and deployment projects. By adopting and adapting such a reference base, enterprise personnel can cooperate in progressing enterprise engineering projects, improving the enterprise, improving stakeholder's communications and utilizing enterprise resources. By adopting appropriate tool sets, enterprise personnel can reuse in a practical way explicit enterprise designs and models to realize the benefits of enterprise engineering on a continual basis and realize further improvements in enterprise operation.

Such a reference base needs to include capabilities that:

- capture concerns of mission fulfilment stakeholders (manufacturing, transport, service delivery, etc.) and of business stakeholders;
- describe suitable solutions to identified problems within the enterprise;
- model the whole life history of an enterprise integration project from its initial concept through development, operation and finally decommissioning or obsolescence; and
- encompass the people, processes, resources and organizations involved in performing, managing, and controlling the enterprise mission.

From an enterprise engineering perspective, the following distinction is drawn:

- enterprise architecture refers to the arrangement of physical components, logical relationships, and human interactions involved in the development, implementation and operation for a programme such as enterprise integration or other enterprise related programme, usually including a set of projects; and
- system architecture refers to the arrangement of physical components and logical relations of a system that is a constituent of an enterprise; for example, the computer-control-system part of an overall enterprise or product.

This document does not present or adopt specific methodologies for creating or using enterprise architectures or models. The focus is on establishing a reference base capable of supporting specific enterprise programmes, rather than a design intended to fulfil the stated requirements.

This document identifies an extensive collection of potential artefacts for expressing an enterprise-referencing architecture and its associated methodologies. Not all of these artefacts will be applicable, necessary or even desirable for all architecting efforts. The identification of these artefacts assures that this document meets the needs of the widest possible number of enterprise-referencing architecture and methodology situations. Users of this document need to assess not only the value of generating an identified artefact but also the value of maintaining that artefact under the changing circumstances of the referenced enterprise.

0.2 Rationale for this document

Well-designed standards in the domain of enterprise integration and modelling provide a point of reference for enterprise architects and designers, thereby significantly reducing the risk of investing in islands of integration. Where an island does exist, these standards assist the architect or designer to create the translation necessary for the island to interact within an established context. A standard for enterprise-referencing models enhances interoperability by establishing the elements that are required in a model intended to support enterprise architecture.

This document defines concepts, rules and requirements for architecture descriptions that refer to an enterprise, most often articulated as models, with the intent to guide and constrain other standards or implementations that do or will exist on the topic. To realize this intent, this document specifies the concepts to use when producing an enterprise-referencing architecture (see [Clause 5](#)) and when constructing enterprise-referencing models (see [Clause 6](#)). This document provides a reference base, guidelines and constraints for enterprise architecture and models to anyone engaged in enterprise activities where models are utilized.

0.3 Benefits of this document

The requirements of the reference base for enterprise-referencing architecture and models in this document allow assessment of an enterprise architecture framework or other approach as well as associated methodology and languages for completeness with respect to current and future purpose of the architecture effort. This document will help guide development programmes.

When conforming implementation designs have the same technology areas and nomenclature, or can map to them readily, the information from one enterprise or process is more readily sharable with information of another enterprise or process.

The benefit will be most relevant to any group charged with improving an enterprise infrastructure or its processes. Such a group will find it necessary to either select or create a reference architecture of its own with terminology that pertains specifically to the company, industry, and culture involved. This document will help guide that selection or creation.

This document expects users that are:

- enterprise planners, builders, modifiers, and analysts using the requirements to check completeness of their activity;
- enterprise-referencing model builders using the requirements to assure consistency between models to enable model interoperability; and
- developers of standards for enterprise representation using the requirements to assure consistency between their standards and this document.

NOTE While the requirements specified herein cover a broad range of enterprise considerations, many users find it advantageous to structure the topics considered into a specific framework for their own work. Such a framework is a methodological choice of the user and goes beyond the scope of this document. [Annex B](#) discusses a framework consistent with this document and other International Standards provide further framework guidance.

STANDARDSISO.COM : Click to view the full PDF of ISO 15704:2019

Enterprise modelling and architecture — Requirements for enterprise-referencing architectures and methodologies

1 Scope

This document specifies a reference base of concepts and principles for enterprise architectures that enable enterprise development, enterprise integration, enterprise interoperability, human understanding and computer processing. This document further specifies requirements for models and languages created for expressing such enterprise architectures.

This document specifies those terms, concepts and principles considered necessary to address stakeholder concerns and to carry out enterprise creation programmes as well as any incremental change projects required by the enterprise throughout the whole life of the enterprise. This document forms the basis by which enterprise architecture and modelling standards can be developed or aligned.

This document does not define standard enterprises, standard organizational structures, standard enterprise processes, or standard enterprise data. In addition, this standard does not specify enterprise modelling processes.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1

architecture

conceptualization of the form, function, and fitness-for-purpose of an *enterprise* (3.4) in its *environment* (3.9), as embodied in the elements of the enterprise, the relationships between those elements, the relationship of the enterprise to its environment and the principles guiding the design and evolution of the enterprise

Note 1 to entry: Since architecture in an enterprise context, e.g. an enterprise architecture, is always conceptual, the expression of architecture as an architecture description is through models of the enterprise reality to which the architecture applies. Detailed designs conforming to an enterprise architecture description often take less abstract forms as they approach enterprise reality.

Note 2 to entry: The notion of fitness-for-purpose distinguishes architecture from other characteristics of design by adding the obligation of elegance in efficiency and effectiveness to design form and function — a distinguishing characteristic separating an ad hoc assemblage of parts necessary to conduct an enterprise from an intentional composition of constituent elements necessary to achieve and sustain superior enterprise performance.

[SOURCE: ISO/IEC/IEEE 42010:2011, 3.2, modified — Content of original definition has been adapted to the context of this document and Notes to entry have been added.]

3.2

aspect

distinguishing characteristic, manifest by content projection from an integrating *enterprise model* (3.6)

Note 1 to entry: In this document the term *aspect* usually refers to a collection of concerns that are better addressed by looking across various *views* (3.23), i.e. concerns best addressed by a view across other model views. The scale and scope of these kinds of broad-based concern collections are poorly handled by a single model representation but can be effectively rendered by selecting a composite view from existing model view content.

3.3

business process

partially ordered, often nested, set of *enterprise* (3.4) activities that can be executed to achieve some desired result in pursuit of a specified objective of an enterprise or a part of an enterprise

3.4

enterprise

human undertaking or venture that has explicit and clearly defined *mission* (3.13), goals, and objectives to offer products or services, or to achieve a desired project outcome or business outcome

Note 1 to entry: When in operation, an enterprise functions as a system that realizes the mission through life cycle activities within a life history.

Note 2 to entry: In this document, “enterprise” refers to concrete (e.g. company, project or extended supply chain enterprise) or abstract (e.g. virtual enterprise) entities.

Note 3 to entry: One or more organizational unit participates in an enterprise. For an enterprise involving more than one such unit, each brings various *resources* (3.21) forward for use, participating to the extent that it benefits from their involvement, which most often occurs to address some challenge that it is unable to address on its own.

3.5

enterprise engineering

discipline applied in carrying out any efforts to establish, modify, or reorganize any *enterprise* (3.4)

Note 1 to entry: An enterprise engineering and architecting methodology provides guidance for efforts to establish, modify, validate or reorganize an enterprise.

3.6

enterprise model

representation of an *enterprise* (3.4) as well as entities within an enterprise, their interrelationships, their decomposition and detailing to the extent necessary to convey what the enterprise intends to accomplish and how it operates

Note 1 to entry: An enterprise model, which is used to improve the effectiveness and efficiency of the enterprise, identifies and specifies essential components and elements to any necessary extent of detail, including any subsystems and constituent models of the enterprise, e.g. an enterprise *architecture* (3.1) model.

[SOURCE: ISO 19439:2006, 3.23, modified — The words “abstraction of an enterprise domain that represents enterprise entities” have been replaced with “representation of an enterprise as well as entities within an enterprise”, the words “what it intends” have been replaced with “what the enterprise intends”, and Note 1 to entry has been added.]

3.7

enterprise-referencing

applicable to an *entity* (3.8) that is, or includes, or is part of an *enterprise* (3.4)

Note 1 to entry: A generic enterprise *architecture* (3.1), an architecture for a specific enterprise, and an architecture that includes an enterprise as one of its elements are all enterprise-referencing architectures.

3.8**entity**

concrete or abstract thing in the domain under consideration

[SOURCE: ISO 19439:2006, 3.29]

3.9**environment**

context that determines the setting and circumstances of technological, business, operational, organizational, political, regulatory, social, and other critical influences and constraints upon an *enterprise* (3.4), which affect or are affected by its development and behaviour, but are not controllable by the enterprise itself

Note 1 to entry: In the case of a nested enterprise, its environment may be within the larger enterprise.

[SOURCE: ISO/IEC/IEEE 42010:2011, 3.8, modified — Content of original definition has been adapted to the context of this document, incorporating content from ISO 19439:2006, 3.30, and Note to entry has been added.]

3.10**framework**

structure expressed in diagrams, text and formal rules which relates the elements of an *enterprise* (3.4) *architecture* (3.1) to each other

Note 1 to entry: The purpose of a framework is to guide creation of one or more enterprise architecture descriptions.

[SOURCE: ISO 19439:2006, 3.31, modified — The words “the components of a conceptual entity” have been replaced with “the elements of an enterprise architecture”, and Note 1 to entry has been added.]

3.11**genericity**

extent to which a concept generalizes entities in a category or group

3.12**life cycle**

set of distinguishable phases and steps within phases which an *entity* (3.8) goes through from its creation until it ceases to exist

Note 1 to entry: The phases of a nested *enterprise* (3.4) can be steps within the life cycle of the containing enterprise.

[SOURCE: ISO 19439:2006, 3.42, modified — Note 1 to entry has been added.]

3.13**life history**

actual, recorded and configuration managed sequence of phases and steps within phases that an *entity* (3.8) goes through during its lifetime

3.14**mission**

characterization of the effect that an *enterprise* (3.4) expects to achieve through the fulfilment of functional requests for products or services

3.15**model**

representation of certain entities and their characteristics either (a) using a formalism, or (b) using an established or ad hoc modelling paradigm, approach, or technique

Note 1 to entry: An ad hoc model is more difficult to implement using an established design technique.

Note 2 to entry: A model may be a subset of a broader model.

3.16

model-based

represented using a formalism which has a formal syntax and semantics, usually with a theoretical basis, and expressible in a symbolic language

Note 1 to entry: Presentation of such models is often graphical but the definition mandates that the graphical representation be translatable into a symbolic language, thereby constraining interpretation of the graphical representation.

Note 2 to entry: In order to satisfy specific *stakeholder* (3.22) concerns, “model-based” is often used as a qualifier to characterize a kind of design, or practice, e.g. model-based system engineering, model-based design, model-based specification.

3.17

modelling dimension

conceptual collection of *enterprise* (3.4) *entity* (3.8) *viewpoints* (3.24) related by kinds of architectural concerns and exhibiting reasonably distinct aggregating coordinates along a continuum

Note 1 to entry: Common modelling dimensions are life cycle, viewpoint, and *genericity* (3.11).

3.18

organization

distribution of responsibilities and authorities in the *enterprise* (3.4)

Note 1 to entry: Within an organization, people are often given role titles designating collections of responsibilities and authorities with roles arranged in an accountability hierarchy.

3.19

perspective

orientation of a *stakeholder* (3.22) or *model* (3.15) user relative to an identified domain

Note 1 to entry: A stakeholder's orientation can be formed by stakeholder concerns, as well as by their training, experience, cultural background, and their motivations.

3.20

reference base

source of information comprising descriptions of generalized *enterprise* (3.4) *architecture* (3.1) concepts, requirements and recommendations

3.21

resource

entity (3.8) that provides some or all of the capabilities required to execute an *enterprise* (3.4) activity

Note 1 to entry: In this document, resource is used in the system theory sense of entities that provide capabilities required by the system and are an essential part of the system itself. The resource description includes the identification and description of consumables (such as energy, air, coolant) that are required to be present in sufficient quantities to operate the resource. In contrast, material is reserved for process inputs that are required by the various activities such as raw materials, parts and assemblies.

Note 2 to entry: In this document, the term “resource” applies to non-human entities involved in enterprise entity operation. The involvement of humans is described by roles they carry out in the enterprise entity operation.

[SOURCE: ISO 19439:2006, 3.60, modified — The word “enterprise” has been deleted at the start of the definition, the last sentence of Note 1 to entry has been removed, and Note 2 to entry has been added.]

3.22

stakeholder

individual, team, organizational unit, or class thereof, having concerns relative to their *perspective* (3.19) about an *enterprise* (3.4), or its *architecture* (3.1), or in an *architecture entity* (3.8) of the enterprise

Note 1 to entry: Typical enterprise stakeholders include enterprise owners, enterprise customers, and enterprise employees responsible for receiving or delivering either products or services, and those persons or *organizations* (3.18) partnering with the enterprise to achieve its mission.

Note 2 to entry: Enterprise architects are stakeholders of the enterprise architecture, but not necessarily stakeholders of the enterprise itself. A user of enterprise architecture or related models is an enterprise stakeholder. There can be observers of an enterprise that are not stakeholders for that enterprise.

[SOURCE: ISO/IEC/IEEE 42010:2011, 3.10, modified — Content of original definition has been adapted to the context of this document and Notes to entry have been added.]

3.23

view

work product expressing a selective perception or representation of a *model* (3.15), which emphasizes some specific feature or characteristic and disregards others

[SOURCE: ISO 19439:2006, 3.25, modified — Content of original definition has been adapted to the context of this document.]

3.24

viewpoint

identification of one or more kind of *model* (3.15) useful for addressing a collection of related *stakeholder* (3.22) concerns

Note 1 to entry: When the term viewpoint is used without any qualifier it refers to the general case. When the term viewpoint is used with a qualifier, the viewpoint applies to a specific collection of concerns, e.g. operational viewpoint, capability viewpoint, services viewpoint.

Note 2 to entry: A viewpoint identifies one or more models necessary and sufficient for expressing one or more *enterprise model* (3.6) *views* (3.23) that address the specific collection of related stakeholder concerns.

4 Abbreviated terms

CIM	Computer Integrated Manufacturing
CIMOSA	Computer Integrated Manufacturing Open Systems Architecture
EA	Enterprise Architecture
EAET	Enterprise Architecture and Engineering Tool
EAM	Enterprise Architecting, Engineering and integration Methodology
EI/EA	Enterprise Integration/Enterprise Architecture
EMEIS	Enterprise Model Execution and Integration Services
EML	Enterprise Modelling Language
EMO	Enterprise Module
EM	Enterprise Model
EOS	Enterprise Operational System
FIRO	Function, Information, Resource and Organization
GEM	GRAI Evolution Method
GEMC	Generic Enterprise Modelling Concept
GERA	Generalized Enterprise Reference Architecture
GERAM	Generalized Enterprise Reference Architecture and Methodology

GIM	GRAI Integrated Methodology
GRAI	Graphs with Results and Actions Inter-related
IT	Information Technology
ODP	Open Distributed Systems
OMG	Object Management Group
PEM	Partial Enterprise Model

5 Requirements for enterprise-referencing architectures and models

5.1 General requirements

The requirements for enterprise-referencing architectures and models are cast as a framework relying upon the key principles of enterprise integration and interoperation found in [Annex A](#).

The framework comprises the requirements for enterprise-referencing architectures in [Clause 5](#) and models with modelling languages and associated methodologies in [Clause 6](#). These architecture and model requirements assist with planning enterprise integration and interoperation by an individual or team, which determines and develops a course of action that is complete, accurate, and properly oriented to future business developments and economy of resources. The planners shall establish the following objectives among other potential objectives:

- a) identify enterprise boundaries and relevant relationships to the environment in which it operates;
- b) define present and possible future enterprise objectives;
- c) describe significant tasks to perform;
- d) identify the necessary kinds and quantity of information;
- e) identify relevant enterprise elements and their relationships to enterprise objectives;
- f) specify relationships among humans, processes, and equipment for the interoperation considered;
- g) specify sufficient management functions and responsibilities;
- h) identify relevant economic, cultural, and technological factors;
- i) describe the extent of automation-support required;
- j) provide modelling that can trace the whole life history of an enterprise entity including entity and relationship evolution;
- k) describe decision-making structures and means for detecting inconsistencies;
- l) maintain both human readable and machine processable forms of expression;
- m) measure sufficiency and efficiency of enterprise architecting methodology as the extent to which the enterprise achieves objectives.

Enterprise-referencing architectures and models can address the role of humans, the description of processes, the identification of information, their relationships, and the representation of all supporting technologies throughout the life cycle of the enterprise.

Effort applied to the development of architectures, models and views needs to remain cognizant of the intended objectives and benefits to the enterprise, and to balance the effort required to achieve these outcomes. Typically, effort will be constrained by cost and/or schedule.

5.2 Applicability and coverage of enterprise architecture

5.2.1 Kinds of enterprise

Enterprise-referencing architectures and the models that represent those architectures should be suitable for supporting the engineering of manufacturing and business enterprises of many conceivable sizes or varieties. Supported enterprise entities may be independent enterprises, e.g. companies, or parts of an enterprise, e.g. divisions, departments, or the business of collaborating and cooperating enterprises, e.g. supply chains, virtual enterprises, or even the enterprise operational systems or products. Reference architectures may exist for specific enterprises or systems, e.g. discrete parts manufacturing, process industries, complex system of systems interactions and behaviours, and information systems. The breadth and depth of the domain covered by an enterprise-referencing architecture and its representational models shall be identified clearly.

5.2.2 Characterizing enterprise architecture

A distinguishing characteristic of enterprise architecture arises because of the nature of enterprise creation and the purpose of the architecture relative to that enterprise. One kind of architecture — system architecture — relates to the design of a system, e.g., the computer control system part of an overall enterprise integration project. Another kind of architecture — enterprise architecture — involves the design of the enterprise that produces that system architecture. Enterprise architecture enables the creation of the system architecture by structuring the development and implementation of a project such as an enterprise integration or other enterprise development programme, which includes the creation of architecture descriptions in support of enterprise products or services.

Any elaboration or derivative of this document as a reference base for enterprise-referencing architecture and models shall be applicable for each enterprise entity within a specified domain.

The concrete form of an enterprise architecture conceptualization shall be as an architecture description, such description being suitable to the purpose for which the conceptualization occurs.

NOTE 1 In this document, the appearance of the term “architecture” often means its concrete form as an architecture description.

NOTE 2 Often one enterprise architecture supports the development and implementation of another enterprise, which produces its own enterprise architecture in support of the development and implementation of a product or service. See [Annex B](#) for more discussion of enterprise and architecture kinds and relationships among enterprise architectures.

5.2.3 Enterprise engineering and architecting methodology

A methodology, using this document or an elaboration or derivative of this document as a reference base, shall provide the necessary guidelines and management techniques for the identification of projects or programmes that support the deployment of an enterprise entity. Such a methodology can be model-based. While the enterprise engineering process can result in a specific enterprise model, the methodology shall be suitable for expressing concepts as described in [5.3](#) and components as described in [Clause 6](#).

Such methodologies shall guide the user in the process of change management and provide methods of progression for every kind of life cycle activity for any enterprise entity within the scope of the reference base. These methodologies shall describe the process of enterprise integration and enterprise modelling that may include conceiving, defining, expressing, documenting, communicating, certifying proper implementation of, maintaining and improving the architecture of an enterprise. Different methodologies can exist to accommodate different characteristics of the enterprise change processes, whether complete integration processes, or incremental changes as experienced in a continuous improvement process.

NOTE Enterprise architectures and models need not rely upon a specific methodology and its accompanying framework. Potentially, many different methodologies and/or frameworks are possible to achieve the desired outcome. The primary consideration is applicability and suitability in relation to these requirements.

5.2.4 Enterprise design

Enterprise architectures and models shall identify the information and activities necessary for managing, conceiving/defining, describing, designing, implementing, operating, maintaining, decommissioning, and disposing of an enterprise, and shall arrange enterprise activities in a form, e.g. a model, that allows a determination of their dynamic behaviour. Enterprise engineering uses the identified elements and interactions to describe in both human and machine-readable form the structure and behaviour of the enterprise.

5.2.5 Enterprise operation

Enterprise architectures and models shall identify the activities necessary for using the results of enterprise engineering in the operation itself. Such use may include model-based decision support and model-driven operation monitoring and control.

5.3 Essential concepts for enterprise-referencing architecture

5.3.1 Span of conceptual orientation

Various enterprise architecture property concepts are necessary to address the range of concerns expressed by stakeholders. Conceptual orientations express requirements for these properties.

Enterprise-referencing architecture concepts shall support enterprise engineering, operation of business processes, facility layout models, information system models, communication system models, logistics models, and models of enterprise resources, e.g. information technology, manufacturing technology, office automation.

5.3.2 Human oriented

Enterprise architectures and models shall identify and provide the means to represent human participation, such as organizational and operational roles, capabilities, skills, know-how, competencies, responsibilities, accountabilities, authorization, and relationships to the organization.

5.3.3 Process oriented

Enterprise architectures and models shall identify and provide the means to represent the enterprise operation. Such representations shall cover the behaviour of the operation including functionality, data flow among functions, and sequencing of control. The representations shall recognize the life cycle ([5.3.9.1](#)) and life-history ([5.3.9.2](#)) concepts and shall support process-oriented operations.

5.3.4 Interoperation oriented

Enterprise architecture and models shall identify and provide the means to represent collaborations and exchanges of entities, both information and physical, with external organizations in the enterprise environment.

5.3.5 Decision oriented

Enterprise architectures and models shall identify and provide the means to represent enterprise decision-making including the decision system structure, decision centres and decision processes.

5.3.6 Realization oriented

5.3.6.1 Mission-fulfilment oriented

Enterprise architectures and models shall identify and provide the means to represent any process, and its constituent activities, involved in realizing the established mission of the enterprise in terms of providing enterprise products and services to its customers.

5.3.6.2 Mission-control oriented

Enterprise architectures and models shall identify and provide the means to represent any process, and its constituent activities, for accomplishing management and control in support of the mission of the enterprise according to the criteria established by enterprise management.

5.3.6.3 Economic oriented

Enterprise architectures and models shall identify and provide the means to represent economic characteristics of the enterprise through means to estimate and calculate the economic parameters of enterprise operation.

5.3.7 Technology oriented

Enterprise architectures and models shall identify and provide the means to represent all technologies employed in the enterprise operation.

5.3.8 Environment oriented

Enterprise architectures and models shall identify and provide the means to represent all concerns related to the interaction between the enterprise and its economic, social/political, and ecological environment, including concerns relating to the impact of creation, operation, transformation and decommissioning of the enterprise or of its constituent parts.

5.3.9 Lifetime oriented

5.3.9.1 Life cycle oriented

Enterprise architectures and models shall identify and represent the life cycle phases that are pertinent during the life of any enterprise. Different life cycle phases may have different models and different models may exist at an individual modelling phase. Where determined necessary, these models shall have the capability to interoperate and communicate with each other.

Life cycle phases shall encompass all activities from inception to decommissioning at the end of life of the enterprise. The characterization of life cycle phases shall be with respect to mutual dependencies and similarity of model content rather than a presumed chronological sequence. Life cycle phases may be concurrent, repeated, or empty as the enterprise modelling situation necessitates.

NOTE 1 If a phase is decomposed, different subsets can have different dependencies and therefore scheduled with more flexibility. This is particularly evident with software, where decomposition, to facilitate modularity, is a desired practice.

NOTE 2 Feeding model information forward and backward in life cycle activities enables value added iteration of enterprise processes that improves product quality. A phase can modify or extend the model of the previous phase according to the evolving needs of the following phase.

NOTE 3 Other standards, particularly those aligned with ISO/IEC/IEEE 15288, use the term stage rather than phase for a similar concept related to the development of a system. To maintain the distinctions between system architecture and enterprise architecture, this document adopts the term phase to characterize segmentation of activities during the lifetime of an enterprise.

5.3.9.2 Life history oriented

The life history shall express the chronology of life cycle phase activities as they occur during the life span of the enterprise. Enterprise architectures shall specify methodologies capable of representing the life history of any enterprise; that is, the representation in time of activities carried out by the enterprise.

5.3.9.3 Complementary role of life cycle and life history

Using the life-oriented concepts of [5.3.9.1](#) and [5.3.9.2](#), the architect identifies enterprise activities within a life cycle phase, while life history allows the architect or user of the architecture to identify the corresponding time period in which the instances of these activities occur. This dual orientation accommodates the iterative nature of the life cycle activity dependencies while maintaining the time sequence of life history. These activity phase iterations identify different change processes required on the operational processes and/or the product development or customer services. See [Figure B.3](#) for an illustrative example.

NOTE 1 Common acquisition practice mandates delivery milestones that strictly sequence the execution of life cycle phases chronologically such that the life history throughout acquisition mirrors completion of life cycle phase deliverables (e.g. the waterfall approach). However, the actual practice of enterprise architecting often necessitates a revisiting of specific phase activities to accommodate revisions and change orders (e.g. the spiral approach and the agile approach), thus life history chronicles the occurrence of all activity iterations as they actually occur in time, while allowing the overall deliverable milestones to remain as selected views of life history.

NOTE 2 Other standards, particularly those aligned with ISO/IEC/IEEE 15288, use the term life cycle stage instead of life history stage.

5.3.10 Stakeholder oriented

The enterprise architecture shall identify all stakeholders for that enterprise and their individual interests or concerns with respect to their perspective regarding the enterprise. Stakeholders and their perspectives can change during the life-history of the enterprise, therefore the iterative performance of life cycle phases can need to address changed stakeholder concerns.

Each enterprise stakeholder perspective shall have one or more concerns related to various characteristics of the enterprise life cycle and any individual concern can be shared by more than one stakeholder's perspective. The enterprise architecture description shall identify the explicit relationship between stakeholders, perspectives and concerns.

The expression of a stakeholder perspective concern relative to the enterprise shall be in terms consistent with those commonly used within the enterprise. Since enterprise concerns are the drivers for enterprise architectures and models, enterprise architectures and models shall clearly identify those enterprise concerns that are captured by reference.

5.3.11 Viewpoint oriented

An enterprise architecture shall provide a means for capturing and expressing collections of concerns, originating from stakeholder perspectives, as viewpoints from which views of the enterprise are made manifest during its lifetime. An enterprise viewpoint should establish the conventions for creating, depicting, interpreting, and analysing one or more model views of the enterprise associated with the viewpoint. Viewpoint conventions shall include the languages, notations, and kind of model for expressing the associated views, and any modelling methods, techniques for analysing, or other operations for the construction of the view.

An enterprise architecture description may use one or more viewpoints. Each viewpoint provides a collective framing of enterprise related concerns expressed from the perspective of stakeholders.

Enterprise architectures and models shall specify a set of viewpoints sufficient for spanning the expressed concerns of relevant stakeholder perspectives.

A viewpoint within an enterprise architecture may specify more than one kind of model, e.g. an information model, a process model, a simulation model, perhaps by application at different life cycle phases. The application of such a viewpoint to the content of those models results in a view from each model consistent with the viewpoint. In some situations, the several views resulting from the application of a single viewpoint need to be composed into a more meaningful representation for stakeholders (see [5.3.13](#) and [B.3.1.5.3](#)).

A viewpoint can result in more than one architecture view, again perhaps one for each kind of model at each life cycle phase. Each view shall be associated with only one instance of the viewpoint's application, i.e. each application produces only one view instance, but there may be many views resulting from repeated instantiation of views by application of that one viewpoint. Similarly, each selection of a parameter value for a parameterized viewpoint may yield only one architecture view for each parameter selection, although that view can have different expression over the lifetime of the enterprise entity.

The enterprise view resulting from a viewpoint shall span the enterprise relative to the concerns that the view captures and expresses.

NOTE A viewpoint is a point of reference for the specification of corresponding views and all views have a corresponding viewpoint that governs view generation.

5.3.12 Model oriented

5.3.12.1 Representing enterprise architecture

Models of enterprise architecture have an abstract quality making the isolation of non-architectural enterprise models difficult. Architecture models, as representations of physical or conceptual entities, should not articulate those characteristics that do not contribute to the success of the enterprise mission.

In addition, since one enterprise may consist of several other enterprise entities (see [6.3.8](#)), part of an enterprise model for one enterprise may be an enterprise architecture model for another enterprise. Therefore, as a reference base, this document considers all models with an enterprise scope as potentially pertinent to enterprise architecture. [Clause 6](#) presents requirements, recommendations, and options for models typically found in architectural representations of an enterprise.

Following the requirements found in ISO/IEC/IEEE 42010, enterprise architecture models shall relate to one or more architecture viewpoints and its associated stakeholder's architectural concerns.

NOTE In the course of enterprise development some enterprise architecture models can transition into enterprise operational models or model-based executable enterprise processes, thus firmly binding the architecture of the enterprise to its operations.

5.3.12.2 Model-based engineering

Model-based enterprise architectures shall identify and provide the means to model an enterprise within the abstract space defined by conceptual modelling dimensions appropriate for the purpose of the architecture. The model-based formalism as applied to a particular model view shall align with one or more modelling dimensions of the enterprise and the same formalism can be useful for more than one dimension.

NOTE See Warfield and Christakis discussion about understanding dimensions^[20].

5.3.12.3 Genericity

Model-based enterprise architectures shall provide the capability for representing generic enterprise elements ([6.4.1](#)), partial enterprise models ([6.4.2](#)), and particular enterprise models ([6.4.3](#)).

5.3.12.4 Model integration

If an enterprise-referencing architecture takes the form of more than one model, then either these models shall be integrated or mechanisms shall be created to maintain their mutual consistency.

5.3.13 Model view oriented

5.3.13.1 Models and views

Enterprise architectures and models shall represent concepts that allow a description of an enterprise as an integrated model, presenting different aspects of concerns to the user as model views. Views should contain a subset of the content of the integrated model in order to concentrate on relevant concerns or aspects as specified in the governing viewpoint. The content of a model may be a model fragment, i.e. subset of the model, or specified portions of an instantiated model.

Different views may highlight certain aspects of an integrated model and hide other aspects. Model views are an important means for verifying stakeholder concerns for completeness, consistency, and integrability of enterprise models. As a projection of model content, views are the model medium used by humans to understand the enterprise.

The concept of view shall be applicable to models for any kind of enterprise across their entire life cycle.

NOTE Implicit in this notion of modelling view is an integrated model from which facts are taken. However, also possible for view projection is a federated model set consistent with a governing viewpoint. If the constituent models provide the necessary linkage between models or model segments, then a satisfactory model view is achievable.

5.3.13.2 Primary aspect for views

Two aspects are of primary importance in order to understand the enterprise: its structure and its behaviour throughout its life cycle. For this document, an information view shall represent the structural aspect of the enterprise as distinct from its resource and organization arrangements, and a function view shall represent the behavioural aspects of the enterprise arising from the structural connections within the enterprise as identified in that information view.

The information aspect is an orderly structured compilation, description, and representation of enterprise information. This information aspect describes the enterprise elements, their structure, relationships, the information to be supplied for function execution, and information describing function results. A physical model may represent portions of an enterprise that take a physical form as a construction from descriptive information, e.g. a scale model built from a blueprint.

The function aspect is a description and representation of activities and processes in the enterprise. This function aspect describes the processing of elements and the concatenation of single processing steps to process chains, reflecting their logical connection and interdependence. The function aspect emphasizes the representation of system behaviour, mutual dependencies, and influences of elements during function execution in the enterprise.

This document does not address the following:

- the manner in which the model expresses the content of these views;
- which content these views should include or not include;
- the necessity of other basic views, e.g. resource view, organization view, economic view, decision view, risk view, etc.

This document states only that there exists a minimum set of model views necessary to present sufficient material to ensure the architectural concerns for completeness, consistency, and integrability of enterprise architecture and models. Determining which views form that minimal set depends upon the specific enterprise context and purpose for addressing architectural concerns of stakeholders.

5.3.14 Enterprise interoperation oriented

5.3.14.1 Support for model interoperability

An enterprise is often large and complex, resulting in many models for many purposes being necessary to describe what is happening at any one time and over time. Even small enterprise entities can need models of different kinds to satisfy distinct viewpoints and aspects of an enterprise.

Stakeholders, especially users, want these many models to be portable. They want to be able to reuse models across applications and not be dependent upon specific application and tool configuration. They want to implement technology that will improve their capability to provide for information from various models to be usable by applications and platforms throughout an enterprise.

Therefore, model representations for architectures in an enterprise context should support interoperation between models within and across organizations within the enterprise environment for both enterprise models and operations. Depending on the situation, there may be different approaches to overcome existing interoperability barriers (see ISO 11354-1).

NOTE As enterprise interactions of systems of systems become more complex, models can have limitations on their ability to predict all behaviours that occur as an aggregation of multiple complex system dynamic states. Research into complex modelling and current limitations is ongoing.

5.3.14.2 Support for process interoperability

Interoperability requires processes to communicate with the environment as well as internally with the processes of other enterprise entities. The extent and variety of successful communication will depend on the skills of the humans observing the processes and the quality of the software employed to enable the communication. Therefore, enterprise-referencing architecture shall provide the means for inter-process communication to the extent necessary to satisfy stakeholder concerns.

NOTE An enterprise-referencing architecture can support capability profiles and collaboration views that identify the communication capabilities and requirements of the processes that have a need for communication.

5.3.15 Verification and validation oriented

Enterprise architectures and models should include the specification of quality assurance and performance criteria that provide a means to assess both the verification of necessary relationships between models at adjacent extent of design or detail, i.e. whether the detail designs conform to the architectural specification, and validation of expected relationships between non-adjacent extent of design or detail, e.g. whether the operations of the enterprise satisfy the stakeholder's concerns.

NOTE Broadly speaking, validation assures the successful alignment of requirements through design to implementation thereby realizing intention and accomplishment of mission.

6 Components of enterprise-referencing architectures

6.1 Enterprise-referencing models

6.1.1 Purpose of enterprise-referencing models

Enterprise-referencing models shall describe essential and relevant characteristics of an enterprise consistent with one or more expressed viewpoints and shall match the extent of elaboration to the extent of the concerns addressed by those viewpoints. Enterprise-referencing models are tools for describing and representing an enterprise in the context of a given purpose. Such models may analyse enterprise operations, guide their engineering and manage their operations. Since enterprises are often analysed and modelled as systems, their models should conform to the relevant elements and principles of the corresponding theory of systems^[63]. Models shall have explicitly stated purpose, assumptions and constraints.

As appropriate, enterprise-referencing models shall address relevant concerns about the enterprise necessary to:

- a) conceive, design, procure, and construct an enterprise consisting of any set of related chosen processes;
- b) simulate the enterprise operation to verify both the model and the operation that the model represents;
- c) manage and operate an enterprise so that it can meet its objectives; and
- d) support an enterprise to modify, redesign, dismantle or reuse itself.

Enterprise-referencing models do not duplicate reality; they are limited approximations of the subset under consideration. Enterprise-referencing models should be as abstract as possible while meeting the needs of the modeller to address stakeholder concerns. Care is necessary to assure that models are consistent with observed or expected reality.

In a specific context, not all information available to an enterprise-referencing architecture modeller is architecturally significant, e.g. the design and implementation is not affected by the architectural specification even though information about that design and implementation is available^[30]. However, the architect should not ignore the choices made during design and implementation that impact fitness-for purpose of the architecture. In fact, an as-is enterprise architecture results from those very choices made in the past and a to-be enterprise architecture often focuses on the need for different choices going forward.

In an operating scenario, the information captured by an enterprise-referencing model shall be available to humans and machines responsible for successful operations.

6.1.2 Kinds of enterprise-referencing models

6.1.2.1 Forms of enterprise-referencing models

An enterprise-referencing model consists of structured modelling elements and rules about permissible relationships between modelling elements, which convey the information that applications use to do their work. An enterprise is often large and complex and therefore has many models for many purposes. Some of these enterprise-referencing models may use model-based formalisms and some may be very informal.

Because of the diversity in model representations of an enterprise, many encoding formats can be present to meet the needs of information generators and information users. Models can be in forms such as organograms, spreadsheets, engineering drawings, process-flow data, computer aided engineering, computer-aided design, and computer-aided manufacturing files, databases, and just plain text.

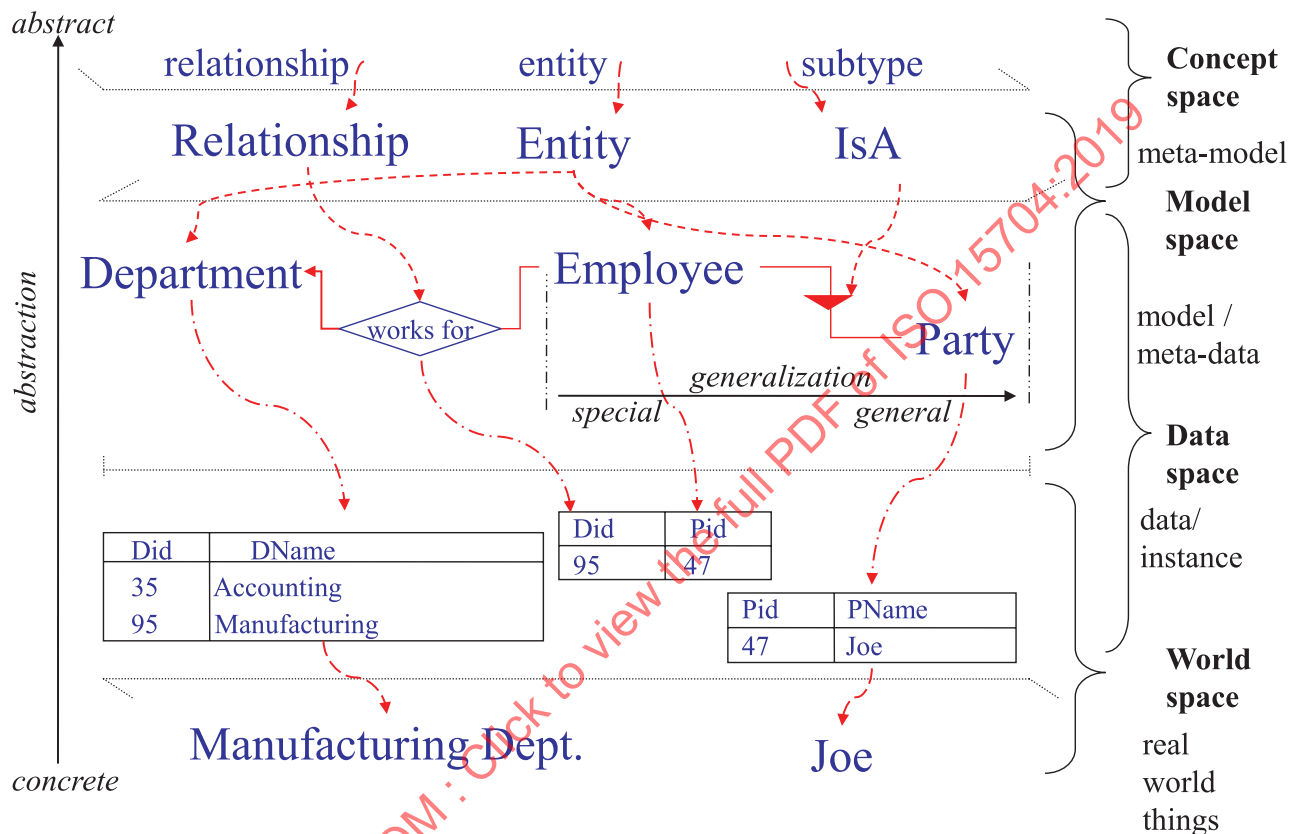
Informal modelling formats make an integrated enterprise-referencing model difficult if not impossible while more formal model-based formats with appropriate model information exchange features provide better integrated enterprise-referencing model opportunities. Such integration is essential to understanding an enterprise in its entirety. As the purpose of an enterprise-referencing architecture moves from narrowly focused stakeholder concerns to encompass a wide range of concerns throughout the whole enterprise, model-based methodologies should become the norm for expressing enterprise-referencing models.

6.1.2.2 Model-based enterprise-referencing models

For model-based enterprise-referencing architectures, the representations at different segments of a life cycle or extent of genericity are often expressed as meta-representations, i.e. the models are distinct abstractions rather than the contents of a generalized enterprise model view as perceived by the model user. The enterprise-referencing model expresses either the means for generating another model more closely associated with the enterprise reality, or the modelling components from which to generate another model more closely associated with the enterprise reality.

Being model-based means that there is a meta-model integral to the methodology that is used to guide the expression of application domain models. Special care is necessary when considering architectures and models with different meta-representations. To the extent possible, an enterprise-referencing architecture should align meta-representations of similar abstraction to facilitate comprehension of the underlying modelled entity^[65].

[Figure 1](#) depicts the meta-relationship abstractions for an employee database using an entity-relationship modelling paradigm.



SOURCE: Martin R. and E. Robertson, “Meta” Matters^[46]. Reproduced with the permission of the authors.

Figure 1 — Meta- relationship abstractions

Sometimes one modelling meta-model satisfies model-based modelling for an entire domain dimension by providing a means for additional model detail, e.g. an information viewpoint across the entire life cycle at all three coordinates of the genericity modelling dimension.

EXAMPLE A management team recognizes that describing product functionality is an important part of enterprise-referencing architecture and identifies basic product features that an architect needs to assess for feasibility within anticipated resource constraints. In this situation, the functional model for the management team is at a different extent of abstraction than the process and resource models created by the architect to assess feasibility.

NOTE A meta-relationship exists between an architectural viewpoint and an architectural model that corresponds to the relationship between the viewpoint and the view of the models governed by that viewpoint. This meta-relationship results from an instantiation of the model to produce the content present in the view.

6.1.2.3 Views of models

A model shall include statements and descriptions of its purpose relative to concerns of appropriate stakeholder perspectives and the constraints imposed upon the modelling effort by the modelling methodology and modeller assumptions. To ensure adequate completeness and consistency with the

governing viewpoint, these artefacts should include a minimum set of derivable model views. Further, because a view is a projection of model content for a specific purpose, an integrated model, i.e. a model composed of other models sharing a common form sufficiently expressive to capture those details that affect interoperability, provides the opportunity for a complete enterprise view rather than a view constrained by lack of interoperability among its disparate models. The number and the kind of model views for inclusion in that minimum set depends upon the methodology used and the purpose and extent of model abstraction.

Consistency is a specific problem when dealing with views for two reasons:

- a) the tendency to use views as an update mechanism;
- b) extent of abstraction among view elements.

Updating through a view does not assure consistency for model content because of unknown effects on the view complement, i.e. model elements not within the view and not subject to the update but changed nonetheless, e.g. the loss of referential integrity when a value changes as can occur across life cycle phase boundaries. A view containing elements of different abstraction extent presents different versions relative to an underlying enterprise reality, e.g. a conflating of data and model within the same view. Enterprise-referencing modelling and operational tools used to manipulate enterprise-referencing architectures and models shall be responsible for ensuring the consistency of model content and fidelity of model views.

NOTE Necessary model views are those that present a useful combination of activities, information, control, resources, and process capabilities to satisfy stakeholder concerns as governed by the corresponding viewpoints. Viewpoint specification in this document includes a call for several explicit architectural views and suggests mandatory compliance with a minimal set of view specifications (see 6.5).

6.2 Modelling languages

6.2.1 Requirements for modelling languages and constructs

Enterprise-referencing architectures that are model-based shall identify modelling languages or modelling constructs that allow description of enterprise operation. Modelling constructs shall allow representation of the different parts of the modelled enterprise, including collaborative interoperation with external partners, and thereby improve both modelling efficiency and model understanding.

The representational form of modelling constructs shall be consistent with the descriptive needs of people creating and using enterprise-referencing models. Therefore, different languages can exist to accommodate the aspects relevant to different model users, e.g. business users, system designers, information technology modelling specialists. In addition, modelling languages can allow the formation of basic constructs into higher order constructs, e.g. macro constructs, to enhance modelling productivity.

NOTE The standardized modelling constructs defined in ISO 19440 improve interchange of models and dialogue between enterprise modellers.

6.2.2 Expressiveness

Enterprise-referencing modelling languages shall be expressive enough to model human roles, operational processes and their functional contents as well as the supporting information, office and production technologies. Their semantics should be describable in terms of ontological theories. This suitability for formal description is especially important if architectural models are to support the enterprise operation itself, because such models need to be executable. However, natural language explanations of concepts should support the definition of the formal semantics embodied in a model as well.

6.2.3 Semantics and syntax of an enterprise-referencing model

Models, as representations of enterprises, shall carry syntax and semantics such that the content of the model is understandable to both humans and machines. The syntax of a model shall refer to the

permissible arrangements of the representations of the elements and to the permissible kinds of relations. The semantics of a model shall encompass the meanings of the elements and relations with respect to enterprise-referencing model concepts. The syntactic form and semantic content of a model may be different depending, for example, on the purpose of the model, its scope and the environment of the enterprise. As machinery takes on a larger share of enterprise operations, properly conveying model intention as well as control become more important.

6.2.4 Names, labels and glossary

Relative to purpose, context, and model constraints, the individual product, process, project, or enterprise will have its own customary names, labels and terminology. To promote understanding about programmes and other co-operative efforts, enterprise-referencing architectures and models shall provide

- a consistent glossary including a syntax and semantics for use in enterprise-engineering and integration efforts, or
- a reference to other suitable glossaries.

Enterprise-referencing architectures and models shall use the glossary content consistently across the enterprise or explicitly state differences in usage.

EXAMPLE [Table 1](#) has three life cycle phase columns with names for each phase. These phase names are intended to describe appropriate constraints on the life cycle content, and enterprise-referencing models are expected to use the given names consistently.

Table 1 — Mapping between system life cycle phases and system activities

Role of Stakeholder	Phase		
	Plan and Build (e.g. before sell/buy/ title transfer)	Use and Operate (e.g. after sell/buy/ title transfer)	Dispose and Recycle (e.g. after product is no longer useful)
Specifies	Develop goals Define strategy Define product needs	Define support needs Define use	Define recycle/dispose needs
Creates	Develop requirements Define concept Design product Plan to produce product	Define use requirements Define support require- ments	Define recycle/dispose requirements
Uses	Procure parts Produce product Test product Ship product	Use the product Support product	Recycle product Dispose product

NOTE A particularly useful set of life cycle phase names are: domain identification, concept definition, requirements definition, design specification, implementation description, domain operation, and decommission definition. See also [Annex B](#) and ISO 19439.

6.2.5 Elements of interoperation

For each model element expressed using a modelling language, the syntax and semantics shall identify:

- form(s) of representation for the element;
- relations between the element and other model elements;

- capabilities of the element;
- dynamics of the interaction involved.

6.3 Models as representations

6.3.1 Representing enterprise characteristics

Various methodologies derived from general systems theory emphasize different characteristics of an entity. The three most frequently utilized characteristics are entity structure, entity behaviour, and entity hierarchy. These characteristics are not mutually exclusive.

Entity structure as used in this document refers to a representation of the internal static or logical structure of the enterprise entity as distinguished from the external structure beyond the entity. Entity behaviour as used in this document refers to a representation of the dynamic responses of the enterprise entity to both internal and external stimuli. The representation of both the structural and behavioural characteristics is often captured through an entity hierarchy of decomposition.

While hierarchy is the primary structural form used to describe relationships among components of the enterprise entity, hierarchy is not the only form used for this purpose. Many enterprises use a matrix form for the allocation of authorities and responsibilities among operational departments. Others use informal or formal networks of expertise to enhance the availability of critical knowledge necessary for enterprise operation. Most often structural forms co-exist within an enterprise, each serving a specific purpose.

NOTE A report from NIST^[52] identifies other characteristics, identified as aspects, of possible relevance: functional, business, human, trustworthiness (security, privacy, safety, reliability, and resilience), timing, data, boundaries, composability, and life cycle.

To adequately represent the architecture of an enterprise, an adopted modelling paradigm should include support for many characteristics of an enterprise, particularly those characteristics of static structure and dynamic behaviour. In addition, a paradigm description shall state the characteristics of an enterprise for which its authors believe it is best suited.

6.3.2 Concepts of internal structure

The structural characteristic has its genesis in the observation that elements are not isolated but have multiple supporting interdependencies with other elements of an entity.

Structure carries both a formal representation and a semantic feature. In most information modelling paradigms, the formal feature refers to the layout of the graph and to the kind of edges. The semantic feature refers to mappings of elements and relations to enterprise-referencing related concepts and beliefs. Mappings may vary widely depending on the purpose of the mapping as well as on the boundary and the environment of the enterprise under consideration.

The formal and semantic distinctions of the modelling approach should be a part of every structural model's documentation, e.g. by reference to a published International Standard or work available to the stakeholders for which the model is created. The representation of structure, particularly an integrated structural model, of an enterprise-referencing architecture should avoid mixing representational approaches.

The kind of internal structuring should be as unambiguous as possible to whatever facility is interpreting the models, either human or machine.

6.3.3 Compatibility of structuring approaches

When alternative approaches to the representation of structure are necessary to properly address stakeholder concerns, any distinctions, both syntactic and semantic, in the representational approaches shall be clearly indicated as part of the model's documentation.

Under some circumstances, models generated using different representational approaches are amenable to federated interoperation. The enterprise-referencing modeller shall ensure that models obtained by the different structuring approaches are able to interoperate where necessary to properly address stakeholder concerns.

6.3.4 Concepts of enterprise-referencing behaviour

The behavioural characteristic relies upon the identification of variables, the dynamic features of an enterprise-referencing entity, and their functional or other relationships. Associated with the behavioural characteristic, to assess or measure the behaviour of the enterprise, an enterprise-referencing architecture should define performance criteria or indicators for attainment of goals, e.g. economic, interoperable capacity. The totality of these performance criteria and indicators with respect to stakeholder expectations shall constitute the fitness-for-purpose of the enterprise-referencing architecture.

NOTE 1 When the variables are restricted to input and output variables, the system is considered as a "black box".

An enterprise is a social hybrid system, determined by properties of humans and machines. Humans in the enterprise, often modelled as objects or role-based resources, have a different behaviour (e.g. learning and problem solving) from machines (e.g. acting and reacting) and sometimes need a different kind of information. Over time, the evolution of machines toward human-like properties will blur this behavioural distinction, making even more important the attention to architecture as a means to characterize appropriate enterprise behaviour and fitness-for-purpose.

Enterprises are dynamic and undergo continuous change due to factors such as changing market conditions, technology, and knowledge. One consequence of changing conditions is the shift from considering the enterprise as hierarchic in both structure and control to considering the enterprise through a distributed or heterarchical characteristic approach where enterprise units communicate and cooperate in both problem solving and action. In turn, this shift makes necessary far greater integration of functions within the enterprise and between enterprises than has occurred in the past. To facilitate this shift, an enterprise-referencing architecture methodology should provide for both hierarchic and heterarchical composition of dynamic elements.

NOTE 2 Models using object-oriented languages and service-oriented architectures for composition of network-based software are two ways in which architecture is supporting a distributed enterprise.

6.3.5 Short-term and long-term behavioural change

An observer can categorize the change in behaviour of a system as either short term or long term. There is no universally recognized distinction between short term and long term, but it is useful to make the distinction at the time of analysis. Criteria used in making the distinction include recognition of change in behaviour, time duration of specific behaviour, and rate of changes in behaviour.

EXAMPLE A drilling machine with a certain drill bit can illustrate behavioural change concepts. The behaviour of the drill bit is of interest to the production-control system as it affects availability of the drilling machine for production purposes. The production-control system categorizes changes in the behaviour of the drill bit as either immediate (short term) or continual (long term). An immediate behavioural change occurs when the drill bit breaks, resulting in a disruption in production until it is replaced with a new bit. A continual behavioural change occurs as the drill bit wears with use. Under usual conditions, the production-control system would plan for a drill bit change at a prescribed time or when the product approaches the minimum quality tolerance. Thus, the distinction made between short-term and long-term change allows the production-control system to specify different behaviours to replace the drill bit.

At whatever enterprise entity extent modelled, the behavioural characteristics shall include recognition of exception to expected behaviour, perhaps as part of an enterprise-referencing risk architecture, as well as the composition of functionalities to achieve expected behaviour over time. As the enterprise entity life cycle unfolds, attention to exceptions is an appropriate means for avoiding disruptions in later phases.

6.3.6 Representation of behaviour

6.3.6.1 Behaviour modelling

Enterprise-referencing models shall have features for describing behaviour; that is, to represent events, actions, conditions, states, state changes, start states, end states, sequencing relationship between actions, and the description of transformation functions including flow of information.

Properties of sequentiality shall be modelled to describe short-term changes whenever the tracing of an individual element is necessary. Enterprise-referencing models used to analyse enterprise performance or to simulate certain processes shall have features for representing effects of sequential phenomena and the time duration of each sequence step. Enterprise-referencing models shall have features for representing time duration, dynamic performance of processes, and sequential phenomena after specific units of time.

6.3.6.2 Time representation

A means for further analysis shall accompany the introduction of the property "time" in terms of duration, e.g., process time. This requirement is a precursor for simulation of behaviour. There are two kinds of behaviour description relative to time: static and dynamic.

6.3.6.3 Static representation

Static representation of behaviour is the description of the relations between elements, e.g., a business process is a logical sequence of relationships between enterprise elements. For this static description, modelling the property time shall not be necessary because it is the potentially allowed sequence of relationships. Time related information is missing, e.g. duration, concurrency.

6.3.6.4 Dynamic representation

Dynamic representation of behaviour provides information about dynamic properties of elements, events and concurrency as well as about the time dependence of elements, attributes and relationships. If the behaviour description is used to simulate the enterprise, the model shall be populated with time relevant information, e.g., starting conditions, capacity, and workload.

6.3.6.5 Sequentiality

Sequentiality is a necessary basis to describe behaviour. Sequential cycles can be considered as similar states being traversed at different times. Measuring sequential cycles in terms of time should enable discrimination between similar cycles that progress at different rates.

NOTE This document uses sequentiality in a broad sense for describing the ordering of activity-associated events, i.e. including serial, parallel, simultaneous, alternative, and repetitive relationships.

6.3.7 Concepts of hierarchy

The hierarchy feature is based on the observation that an element of a system can itself be regarded as a system, which is identified as a sub-system. Similarly, the system under consideration can be regarded as an element of another system, which is identified as a super-system. These relationships imply an assignment to different abstractions or more likely different detail stratum for system elements. Because of interdependence, new properties can emerge at a higher stratum in the hierarchy. In the realm of enterprises, an enterprise entity can subsume other sub-enterprise entities or be subsumed by another enterprise entity.

Each stratum is describable in terms of structure and behaviour. Depending on the desired purpose, specific methodologies are applicable. Steps down the hierarchy expose the inner structure of the sub-system. In the case of an enterprise under development, observation, logical conclusion, or design can provide the exposure. Steps up the hierarchy expose the behaviour of an enterprise in its environment.

Enterprise-referencing architecture with a hierarchy feature shall identify the means by which the exposure of both structure and behaviour occur for each step in a hierarchy.

NOTE Hierarchy is a principle by which the ranking and sorting of real world items and abstractions occurs. There are two kinds of hierarchies. Part-of hierarchies represent the composition/aggregation of elements or the decomposition/disassembly of an enterprise or other system. Kind-of hierarchies represent the sorting of entities into categories distinguished by either abstraction/concreteness or generalization/specialization, e.g. set membership, type and subtype. Within models, the kind-of hierarchies classify building blocks for entities to be modelled, while part-of hierarchies link models of different scope and detailing granularity.

6.3.8 Recursion in decomposition

The output of an enterprise entity can be a product, which can itself be another enterprise-referencing model, or it can be the strategic-definition entity planning an enterprise entity to produce other enterprise entities or enterprise-referencing models. Each of these enterprise entities can be occurring simultaneously and each occurring enterprise entity should have architectural representations of its structure, behaviour, and fitness-for-purpose.

EXAMPLE 1 As a part of its operations, an Architectural, Engineering, and Construction (AEC) firm has undertaken a project to design a new factory for its customer. The AEC company enterprise (E1) is in the operating phase and has been for several years. The project (E2), which consists of several corporate entities (E1, E3, E4), has an enterprise-referencing model larger in scope than the AEC company. The project is an enterprise (E5) and its model has progressed from concept phase (request for proposal and preliminary design) to operation (after contract award). The new enterprise (E6), and its enterprise-referencing model, are in the plan/build phase until the use/operate phase commences.

Stakeholders can be roughly grouped into three role categories: those whose principal concerns lie with specifying (S) enterprise entity capabilities, those whose principal concerns lie with creating (C) enterprise entity capacity to provide the capabilities, and those whose principal concerns lie with using (U) the capability to provide products and services. Specific enterprise entity activities, S, C, and U, address the concerns within each of these roles. Each of these activities can be further decomposed into another set of similarly defined categories S, C, and U, with corresponding sub-activities (see [Figure 2](#)).

This successive utilization of the same categories is a form of structural recursion that applies an architectural model, as a pattern, to each extent of decomposition. This recursive pattern results in the iteration of activities within a recursive structural decomposition, i.e. each extent of decomposition considers the same structure, albeit the iteration of activities can take different forms and use different activities^[43].

NOTE Different kinds of enterprise-referencing models can represent sub-activities, whether aligned with a concern role or not. Where determined necessary, different kinds of enterprise-referencing sub-activities need to have the capability to interoperate and communicate with each other.

EXAMPLE 2 In a manufacturing enterprise entity, the activity “Produce” can be, in turn, separated into lower-level activities S, C, and U. Activity S is user-needs driven and comprises any activities finally resulting in a request for what to produce. Activity C is technology requirements driven and comprises any activities finally resulting in how to produce the product/system in terms of a release statement. Activity U is task driven and comprises any activities finally resulting in the shipment of the product. Each kind of activity at each level of decomposition can provide feedback to a higher level.

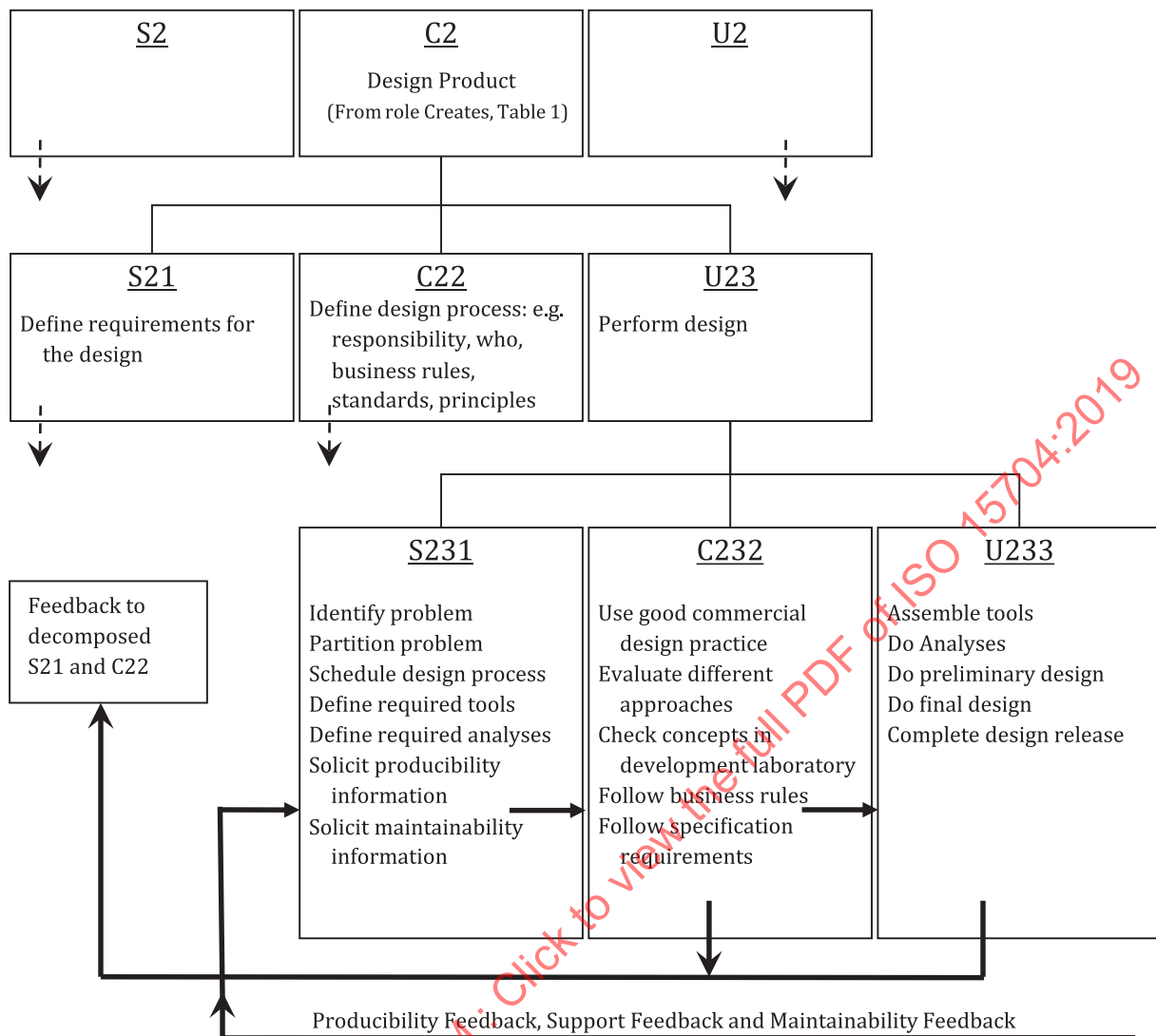


Figure 2 — Decompose Design Product activity to show recursive structure of activities

6.3.9 Iteration

In the [Figure 2](#) example, activities S, C and U are iterative in that they appear repeatedly at each decomposition and across decompositions. Therefore, while shown in the figures as ordered, the activities are not necessarily sequential. The architect may return to previous activities to repeat them with updated input as necessary to satisfy stakeholder concerns, i.e. Producibility Feedback, Support Feedback, and Maintainability Feedback may arise from each decomposition as shown in [Figure 2](#).

Each performance of each activity may result in a different model. Every one of these different models shall be subject to both change and version management. While recursion is structural in this context, iteration is behavioural.

6.3.10 Availability and format of model information

In an operating scenario the relevant information captured by an enterprise-referencing model shall be available to humans or machines responsible for successful operations. The information shall be either in a neutral format or as specified by the using application.

6.3.11 Management of constituent parts

The design of enterprise-referencing models shall occur in a manner that allows the management of their constituent parts by an automated configuration-management system.

6.4 Impact of genericity

6.4.1 Generic enterprise elements

Enterprise-referencing architectures and models may utilize generic elements of enterprise design and modelling. Such generic elements are, in increasing order of formality, glossaries, taxonomies, meta-models, and ontological theories. These elements provide for consistency of enterprise entity representations.

NOTE 1 A formal ontology is considered to be a theory because it states constraints and permitted relationships while a glossary simply states terms and definitions. A taxonomy is a form that expresses relationships among term-based elements but generally lacks explicit constraints. A meta-model expresses the ways in which elements can be used and an ontological theory adds inference capabilities for a network of elements.

Two kinds of generic elements exist for use in developing architectures and models. Abstract elements shall provide a basis for the further expression of distinct detail elements to achieve a concrete architecture or model instance in which the abstraction no longer exists. Generalized elements shall provide a basis for the further expression of architecture or model by the addition of detail while preserving the attributes of the generalized element. For both kinds of generic elements, a gradient of elaboration often exists from the generic element to the realized element instance. For both a generalized element and an abstract element, the realized elements populate the specific enterprise-referencing architecture or model. An architecture model methodology shall make such gradients explicit to enable an observer to identify the origin of constituent parts of the architecture and model descriptions.

NOTE 2 Abstraction gradients are sometimes confused with decomposition hierarchies. The distinction lies in the fact that an abstraction gradient exposes the semantic underpinnings for a transformation from abstract to concrete while a decomposition hierarchy expresses the constituents of its top node.

6.4.2 Partial enterprise models

Enterprise-referencing architectures that are model-based shall support the concept of partial enterprise-referencing models, e.g. reusable reference models. Partial models can range from those targeting industry segments, like automotive or refinery operations, to those targeting common business processes, like labour allocation or order tracking. Partial reference models allow the user to capture and reuse concepts common to many enterprises and thereby to increase the efficiency of architecture model expression and the rate of industry adoption. Partial models still need adaptation to the requirements of the specific enterprise. Partial models may cover one or all of the concepts identified in [5.3](#).

6.4.3 Particular enterprise models

Enterprise-referencing architectures that are model-based shall support the creation of particular enterprise-referencing models that describe part or all of any proposed or actual enterprise.

Enterprise engineering tools can utilize enterprise modelling languages and can provide for the maintaining, creating, analysing, storing, and distributing of enterprise-referencing models. Information technology integration services may support both model creation and model use. The use of such services will ensure access to real-time information in the engineering and operational environments of the enterprise.

6.5 Enterprise perspectives and viewpoints

6.5.1 Primary concern perspectives

6.5.1.1 Identifying perspectives

Sub-clause 6.3.8 identified three role categories for stakeholders: those whose principal concerns lie with specifying enterprise capabilities, those whose principal concerns lie with creating enterprise capacity to provide the capabilities, and those whose principal concerns lie with using the capability to provide products and services. Stakeholders with one of those roles, and potentially stakeholders with more than one of those roles, will articulate concerns pertaining to the role in which they find themselves, i.e. the stakeholder's perspective.

Some concerns can be role specific, but many others can appear as a concern for more than one role, particularly when considering the concerns across the breadth of life cycle phases. Consequently, role-based perspectives alone are unlikely to provide sufficient discrimination for an integrated model that describes an enterprise-referencing architecture. However, perspectives are an effective discriminator for concerns relative to the purpose of the architecture effort, allowing similar concerns from different perspectives an alternative disposition.

The purpose of the enterprise-referencing architecture effort shall serve as a filter for stakeholder concerns, removing from consideration those concerns deemed not relevant to the enterprise mission relative to that architecture effort's purpose. Thus, the purpose of the enterprise-referencing architecture effort shall be the primary determinant of perspectives pertinent for specifying appropriate kinds of models and the resulting views that constitute the enterprise-referencing architecture description. From among those concerns deemed relevant to effort purpose, an architect shall identify collections of related concerns as the basis for articulating one or more viewpoints to drive development of the architecture specification effort. An individual concern may be appropriate for more than one viewpoint.

For each viewpoint in the set of articulated viewpoints, an architect shall identify one or more kind of modelling methodology suitable for conceptualizing ways to harmonize the encapsulated concerns, to represent that concern harmonization, and to express potential solutions to the problem identified by those concerns. From the models generated in response to the viewpoint specification, one or more projection, an aspect view of the model content, should serve to address the concerns within that viewpoint. Where the same kind of model is used to represent more than one viewpoint, consolidation into a single integrated model is useful to assure consistency and completeness of representation. Where different kinds of models are used to represent one viewpoint, composition into a single view is desirable but can be difficult to achieve.

NOTE The phrase "architecture style" captures the distinction between representational purpose of the modelling effort and the purpose or mission of the entity being modelled^[37].

6.5.1.2 Typical viewpoints

This document identifies several viewpoints determined by years of experience to be essential for articulating a holistic integrated enterprise-referencing architecture specification, i.e. they constitute the coordinates of a holistic perspective dimension of the enterprise-referencing architecture. This holistic perspective drives the collecting of various related concerns, i.e. viewpoints, with a set of enterprise entity attributes focusing on one or more characteristics of the enterprise. As appropriate, this holistic perspective applies to concerns associated with each life cycle phase of the enterprise. However, a specific concern should reside within one viewpoint whenever possible to provide clarity for stakeholders.

6.5.1.3 Function viewpoint

Concerns related to enterprise function allow the architect to generate function viewpoints. Corresponding views shall describe and represent the activities and processes of the enterprise. A function view, resulting from the projection of content from a model generated using a kind of model

or models specified by the corresponding viewpoint, should describe the processing of elements and the concatenation of single processing steps into process chains, reflecting their logical connection and interdependence. A function view should emphasize the representation of enterprise behaviour, required capabilities, mutual dependencies, and influences of elements during function execution of the enterprise (see ISO 19440).

6.5.1.4 Information viewpoint

Concerns related to enterprise information allow the architect to generate information viewpoints. Corresponding views shall be an orderly structured compilation, description, and representation of enterprise entity information. An information view, resulting from the projection of content from a model generated using a kind of model or models specified by the corresponding viewpoint, should describe the system elements, their structure, relationships, the information necessary for function execution, and information describing function results (see ISO 19440).

6.5.1.5 Resource viewpoint

Concerns related to enterprise resources allow the architect to generate resource viewpoints. Corresponding views shall describe and represent enterprise entity assets, both technological components and human role capabilities, as utilized in the course of enterprise operations. A resource view, resulting from the projection of content from a model generated using a kind of model or models specified by the corresponding viewpoint, should describe resource instances that align with enterprise activities according to required capabilities. Resource models may structure resources for applications such as asset management (see ISO 19440).

6.5.1.6 Organization viewpoint

Concerns related to enterprise organization allow the architect to generate organization viewpoints. Corresponding views shall describe and represent the responsibilities and authorities within the enterprise domain. An organization view, resulting from the projection of content from a model generated using a kind of model or models specified by the corresponding viewpoint, should gather and structure the different responsibilities for processes, material, information, resource and control of the enterprise, including the mapping of those responsibilities onto the organizational entities and/or organizational groupings such as departments, divisions and sections. An organization view may also provide a representation of responsibilities for decisional activities into a decisional structure for the verification of consistency and completeness (see ISO 19440).

NOTE See [6.5.1.8](#) for an alternative or related decision viewpoint.

6.5.1.7 Economic viewpoint

Concerns related to enterprise economics allow the architect to generate economic viewpoints. Corresponding views shall describe and represent economic components, e.g. influence factors and scalar indices derived from other viewpoints, and analytical methods to inform economic decision-makers. An economic view, resulting from the projection of content from a model generated using a kind of model or models specified by the corresponding viewpoint, establishes the relations between the economic target and the engineering project and is most critical early in the life cycle, when most economic commitments occur, and late in the life cycle, when the assessment of economic performance occurs^[32].

6.5.1.8 Decision viewpoint

Concerns related to enterprise decisions allow the architect to generate decision viewpoints. Corresponding views shall describe and represent the decision centres and decision linkages for planning and control decisions. A decision view, resulting from the projection of content from a model

generated using a kind of model or models specified by the corresponding viewpoint, shall identify the necessary relationships between decision centres.

EXAMPLE In a manufacturing context, those relationships are those that are responsible for getting the correct raw material/product at the correct time, on the correct machine, processed by the correct operator while making decisions across multiple time horizons.

The concerns of a decision viewpoint are the enterprise decision-making structure that provides for identification of decision topics, their categories, criteria and dependencies (see CEN/TS 14818).

6.5.1.9 Risk viewpoint

Concerns related to enterprise risk allow the architect to generate risk viewpoints. Corresponding views shall describe and represent sources of risk to the enterprise and the means and mechanisms for the elimination of risk or the mitigation of the impacts of such risk. A risk view resulting from the projection of content from a model generated using a kind of model specified by the corresponding viewpoint addresses the concerns related to risk, including security, privacy, safety, reliability and resilience. While risk is identified as a distinct viewpoint, risk-based concerns are pervasive in an enterprise, requiring constant consideration both during development and operation^[53].

6.5.1.10 Architecting viewpoint

Each of the concern-based viewpoints previously presented in 6.5.1.3 through 6.5.1.9 has architectural features that contribute to an architecting viewpoint, i.e. a viewpoint governing certain characteristics of architecture description development derived from concerns driven by the architect's perspective. A viewpoint for enterprise architecting informs the selection of appropriate concerns and kinds of models available to the other viewpoints and establishes the conventions by which a model view is created, depicted, interpreted and analysed. Architecting viewpoint conventions include the languages, notations, and kinds of model utilized for producing views of an enterprise architecture, and any modelling methods, analysis techniques, or other operations pertaining to the view creation.

NOTE Concerns expressed by the architecture viewpoint usually are present in other viewpoints as well because those other viewpoints are crafted by the architect using established concerns of stakeholders as well as their own.

6.5.2 Additional concern perspectives

Model developers may generate additional perspective-based viewpoints and corresponding views for other collections of stakeholder concerns. Similarly, enterprise architects may identify additional viewpoints necessary to enable the expression of the concerns of other stakeholders. For some perspectives, a collection of concerns can yield viewpoints satisfied by projections across existing views (see B.3.1.5.3). Where views from many kinds of model are necessary to satisfy a viewpoint specification, e.g. an information model and graphical representation model of that information and a simulation model for monitoring information change, these multi-model artefacts label the resulting view composition as an aspect of the enterprise. Thus, an enterprise aspect may consist of one or more related model views.

For many stakeholders, particularly those with senior management perspectives, these multi-model aspect views are the most meaningful representation of the enterprise. An enterprise modelling tool shall provide the means and methodology for composing multi-model aspect views to the extent that such aspect views are possible.

6.6 Enterprise-referencing modelling framework

A variety of forms and varying extents of detail, often with many inter-relationships and dependencies, characterize enterprise-referencing models, in particular, those models that convey enterprise-referencing architecture. An enterprise-referencing modelling framework shall be a structured set of categorical containers, which provide a means for arranging these many models and associated material according to pre-determined generic viewpoints^[47]. These viewpoints are usually collected

into related groups or modelling dimensions that are arranged for a specific purpose. Such a framework is an architecture for expressing enterprise entity structure, function and fitness-for-purpose.

EXAMPLE The abstract space defined by the dimensions of life cycle, genericity, and modelling viewpoints forms a framework detailed in ISO 19439.

While a framework can structure some of the inter-relationships between the models and model elements that it contains, there will be connections between models and model elements that are not conveyed by the framework structure. An enterprise-referencing modelling framework shall explicitly identify those relationships that it intends to arrange within the containment structure.

Because the modelling framework is a meta-model with respect to the models it contains, the framework may be applied recursively to its model contents to provide further elaboration or decomposition of model components^[43].

6.7 Tools

To aid the architect or modeller in enterprise engineering and integration projects, computer-based tools should support the development and management of enterprise-referencing architectures and models. Such tools shall adhere to one or more enterprise-engineering methodologies with implementation in one or more modelling languages.

Tools shall ensure the maintenance of consistency between model elements during model deployment and refinement processes. Tools shall also provide analysis and simulation capabilities for the creation, manipulation, use, and management of enterprise entity specifications and models, as well as their analysis, description, and evaluation. These capabilities are necessary for decision-making in the course of enterprise engineering. In addition, such tools may support collaborative work across enterprise entity boundaries.

Engineering tools should enable the architect or modeller to connect enterprise specifications and models with the real business process, so as to keep the design specifications and models aligned and up-to-date while maintaining model consistency.

6.8 Modules

Enterprise-referencing architectures and models shall provide the capability for representing the concepts of enterprise reference modules or implementable building blocks or systems or products, or families of products for use as common methodological resources in enterprise entity engineering and enterprise entity integration. One set of enterprise modules important to enterprise entity engineering and integration is the integrating infrastructure or the set of integration-technology services required for enterprise entity engineering and operation in heterogeneous environments.

6.9 Enterprise operational systems

One result of the enterprise engineering process shall be a specification or model for the enterprise operational system. The enterprise operational system shall consist of the humans, hardware and software necessary to fulfil the enterprise objectives and goals in support of the enterprise mission. The content of the enterprise is the result of satisfying the concerns of stakeholders by providing appropriate capabilities either statically or on-the-fly by dynamic matching of required and available capabilities.

6.10 Representation

Enterprise-referencing architectures and models shall provide guidance in the use of the associated components of [Clause 6](#), e.g. a framework or high-level, graphical interpretation. That framework or graphical form should show the applicability of, and relations between, the different components.

Annex A

(informative)

Key principles of enterprise integration and interoperation

A.1 General

Several concepts that describe the nature of enterprise-referencing architectures and models have emerged from the studies of the International Federation of Automatic Control/International Federation for Information Processing (IFAC/IFIP) Task Force on Architectures for Enterprise Integration, which can greatly simplify, integrate, and extend the work of enterprise engineering. This work resulted in the development of the Generalized Enterprise Reference Architecture and Methodology (GERAM), which can support those who plan, design, and implement complex enterprise-integration projects (see [Annex B](#) for an updated version).

Key principles of enterprise architecture are described below to provide a rationale for the requirements in [Clauses 5](#) and [6](#).

A.2 Applicability to any enterprise

The early work in computer integrated manufacturing (CIM) and enterprise integration was confined largely to the field of discrete-parts manufacturing, and to computers and information handling. However, the basic principles involved in enterprise integration apply to any enterprise, regardless of its size and mission or any other such attributes and to all aspects of the enterprise. In addition, it has been a mistake to confine the integration discussions to information and control systems alone. Often problems exist within the mission-control or mission-fulfilment functions, or in manufacturing or other customer product and service operations, or in the associated human and organizational area whose solution would greatly ease the overall system problem, i.e. a total solution needs to involve information, culture, and mission.

Therefore, enterprise programmes should go beyond information and control systems to encompass culture, mission, vision, value, goal, as well.

The reference base for enterprise architecture can be extended to cover all possible enterprises by considering manufacturing as a kind of customer service, providing concept, development, design, modification, production, and supply of goods to the customer. Thus, the mission-fulfilment area of the enterprise architecture would represent the customer service rendered by any enterprise even if that service involved only the supply of information products to the customer.

A.3 Enterprise identification and mission definition

No enterprise can exist in the long term without a business or mission, i.e. it needs to produce products or services desired by its customers. The enterprise usually produces these products or services in competition with other enterprises. Therefore, the enterprise identification and mission definition are essential parts of any enterprise-integration programme.

A.4 Separation of mission-fulfilment functions from mission-control functions

As described below, only two basic classes of functions are involved in operating any enterprise.

One class comprises functions involved in fulfilling the mission, i.e. operating the processes that produce the product or service. In the manufacturing plant these include all material and energy transformation tasks and the movement and storage of materials, energy, goods-in-process, and products and services.

The other class comprises functions involved in the management and control of mission-fulfilment to achieve the desired economic or other gains that assure the viability and continued successful existence of the enterprise. These functions include the collection, storage, and use or transformations of information to control the business processes, i.e. to develop and apply necessary changes to the business processes to achieve and maintain their desired operation. Control includes all planning, scheduling, control, data management, and related functions.

A.5 Identification of process structures

Enterprise operation consists of many transformations of material, energy, and information that can be categorized into two distinct classes: one for information transformations and the other for material and energy transformations. Many separate activities carry out these transformations that can be executed both concurrently and sequentially to constitute processes of an equivalent class. Processes of both classes interface with each other in those activities that request and report status, and in those activities that deliver operational commands. In combination these transformations define the total functionality of the enterprise being considered.

A.6 Identification of process contents

For many technical, economic, and social reasons, humans are involved in the implementation and execution of many business processes of all kinds in mission fulfilment and in management and control. Other processes may be automated or mechanized. The only three classes of implementation tasks or business processes are

- information and control activities that can be automated by computers or other control devices,
- mission activities that can be automated by the mission-fulfilment equipment, and
- non-automatable activities carried out by humans, whether of the information and control or mission-fulfilment class.

A simple way of showing where and how the human fits in the enterprise and how the distribution of functions between humans and machines are accomplished is highly desirable.

A.7 Recognition of life cycle

All enterprises, of whatever kind, follow a life cycle from their initial concept in the mind of an entrepreneur through a series of phases comprising their development, design, construction, operation and maintenance, refurbishment or obsolescence, and final disposal. This partitioning of the life cycle into phases is often considered specific to purpose. However, specific designations neither negate the existence of fundamental transitions through a life cycle nor eliminate a life cycle segment. Phases of this life cycle are not necessarily sequential or linear in appearance. The life cycle principle applies to the enterprise as a whole and to all its constituent parts.

Not only does this life cycle apply to the enterprise but also to the enterprise products as well. Carried further, one enterprise can be the product of another. For example, a construction enterprise can build a manufacturing plant, an enterprise, as its product. The manufacturing plant would then produce its own product, such as an automobile. The automobile also has its own life cycle that goes through similar steps to those discussed here.

The creation and modification of enterprise entities, e.g. development, design, construction, etc., and their use, e.g. operation, are an essential distinction between life cycle phases. This distinction enables the orderly move or release from the engineering environment to the operation environment, providing for verification, testing and release of engineering results prior to operation.

A.8 Enterprise integration

A.8.1 Evolutionary approach to enterprise integration

The integration of all the informational and customer product and service functions of an enterprise can be a part of a master plan. The actual implementation of such integration may be broken up into a series of coordinated programmes that are within the financial, physical, and technical capabilities of the enterprise. These programmes can be carried out individually or collectively, as resources of the enterprise allow. A master plan is useful for scheduling and coordinating programmes and activities.

A.8.2 Approaches to integration

Within an enterprise, the implementation of integration programmes may occur as organization wide processes, or through consistent decision-making, or through consistent data/information, or some combination of these mechanisms.

A.8.3 Extent of integration

Within an enterprise, the implementation of integration programmes may occur between physical entities, application processes associated with enterprise sub-units, or business processes of the enterprise.

Generally, “being integrated” means that material and information flow is timely, accurate and uninterrupted across all mission fulfilment as well as management and control functions, whenever and wherever necessary, and is devoid of unnecessary transformations.

Achievement of integration occurs either statically during design or implementation, i.e. integration by design that is sometimes referred to as design-time integration, or dynamically during operation, i.e. integration through on-demand configuration. The condition for dynamic integration is the presence of self-configuring management and control functions available in the entity by design, and dynamically configurable parts that can interoperate. In practice, the typical scenario is a combination of the two ways to achieve integration, and a task of management is to determine what balance of these two ways to integrate meet the needs for an effective master planning/integration programme and its projects.

A growing trend in enterprise architecture is to favour dynamic configuration through the use of service orientation within the enterprise, whereupon a main actor entity, e.g. a company or a network of companies, develops preparedness to dynamically configure or re-configure virtual enterprises on demand out of the interoperable services of qualified participants.

A.8.4 Assessment and measurement of integration performance

Monitoring by specific measures of mission effectiveness and measures of system performance is essential to all enterprise integration programme implementations. The deviation from expected values for either effectiveness or performance initiates an assessment of the implementation to achieve the objective(s) for which it was undertaken.

A.9 Modularity

Because of the massive nature of most enterprise integration programmes, modularity should occur whenever possible. Thus, it would be helpful if all activities were defined in a modular fashion, along with their required interconnections, so they can later be interchanged with other activities that carry out similar functions but in a different manner should this be desirable. Likewise, these replacement activities would also be best implemented in a modular fashion, permitting their later substitution by still other different methods of carrying out the same function. The choice of these implementation methods can be governed by independent design and optimization techniques if the activity specifications are honoured.

Provided the modular implementation just stated is used, the interconnections between these modules can be considered interfaces. If these interfaces are specified and implemented using company,

industry, and national and/or internationally agreed upon standards, the interchange and substitution noted above will be greatly facilitated.

A.10 Stakeholder concerns

Stakeholder concerns arise from the stakeholder's individual perspective relative to the enterprise. A viewpoint expresses the kinds of enterprise models that can provide views relative to those concerns. Such concerns become the anchor point of enterprise architecture development as stakeholder viewpoints are a starting point to resolve those concerns.

STANDARDSISO.COM : Click to view the full PDF of ISO 15704:2019

Annex B (informative)

Generalized Enterprise Reference Architecture and Methodology (GERAM)

B.1 Introduction

B.1.1 Background

One of the most important characteristics of today's enterprises is that they are facing a rapidly changing environment and can no longer make predictable long-term provisions. To adapt to this change, enterprises themselves need to evolve so that change and adaptation are a natural dynamic state rather than something occasionally forced onto the enterprise. This necessitates the integration of the enterprise operation and the development of a discipline that organizes all knowledge that is needed to identify the need for change in enterprises and to carry out that change expediently and professionally. This discipline is called Enterprise Architecture (EA), which in turn relies on multiple engineering and management sub-disciplines.

Early research, carried out by the European CIM Architecture (AMICE) Consortium on CIMOSA^[36], by the GRAI Laboratory on GRAI and GIM^[34], and by the Purdue Consortium on PERA^[71], as well as several others, produced reference architectures that were intended to structure all enterprise integration knowledge and serve as a guide in enterprise integration programmes. The IFAC/IFIP Task Force on Architectures for Enterprise Integration analysed these architectures and concluded that even if there were some overlaps, none of the existing reference architectures subsumed the others; each of them had something unique to offer. The recognition of the need to define a generalized architecture is the outcome of the work of the Task Force.

Starting from the evaluation of existing enterprise integration architectures [CIMOSA, GRAI/GIM and Purdue Enterprise Reference Architecture (PERA)], the IFAC/IFIP Task Force on Architectures for Enterprise Integration developed an overall definition of a generalized architecture. The proposed framework was entitled the Generalized Enterprise Reference Architecture and Methodology (GERAM)¹⁾. GERAM deals with those methods, models and tools that are needed to build and maintain the integrated enterprise, be it a part of an enterprise, a single enterprise or a network of enterprises (virtual enterprise or extended enterprise).

Independently from this undertaking, with its predominantly manufacturing and control engineering sector influence, other reference architectures [or enterprise architecture (EA) frameworks] developed with initial focus on the development of an IT system. Some of these were extended into the scope of system engineering for the business and defence sectors, e.g. Zachman^[72], Command, Control, Communications, Computer Intelligence, Surveillance, and Reconnaissance/Department of Defense Architecture Framework (C4ISR/DoDAF)^[33], Federal Enterprise Architecture Framework (FEAF)^[38], The Open Group Architecture Framework (TOGAF)^[67], StairLike CIM^[31], and more recently the NATO Architecture Framework^[51], and many others, including proprietary ones, and adaptations or combinations of these reference architectures.

1) The content of this annex is a permitted derivative of GERAM v1.6.3 developed by the IFAC/IFIP Task Force on Architectures for Enterprise Integration. This annex reflects updates and clarifications in GERAM, as well as terminological changes necessary to enable a seamless use in conjunction with systems and software engineering architecture related standards. The term “enterprise-referencing” is now used instead of “enterprise-reference”, in order to indicate that requirements are not limited to a reference architecture for an enterprise. A reference architecture is a subset of the scope for application of the GERAM requirements. See also [B.3.1.5](#) on reference architecture models.

Many of these frameworks have been retrospectively mapped onto GERAM^{[28][55][60][61][44]} in order to identify in a comparable manner their contributions, or potential incompleteness, as well as to determine where those frameworks include additional detail deemed necessary in various typical practical situations.

Compared to the frameworks mentioned above, GERAM is a lightweight framework. For example, GERAM only identifies the need for enterprise engineering or enterprise architecture methodologies but does not prescribe a single one-size-fits-all methodology. Practitioner communities can then develop or adopt a methodology that is fit for the purpose of their context of use, and make it part of the specific EA Framework.

GERAM defines a tool-kit of concepts for designing and maintaining enterprises for their entire life history. GERAM is intended to structure existing enterprise integration knowledge for use when describing architectures involving an enterprise. The framework has the potential for application to all types of enterprise. Previously published reference architectures can keep their own identity, while identifying through GERAM their overlaps and complementing benefits compared to others.

B.1.2 Scope of GERAM

The scope of GERAM encompasses all knowledge needed for enterprise engineering/integration. Thus, GERAM is defined through a pragmatic approach providing a generalized framework for describing the components needed in all types of enterprise integration processes, such as

- major efforts for green field installation, complete re-engineering, merger, reorganization, formation of virtual enterprise or consortium, value chain or supply chain integration, etc., and
- incremental changes of various kinds for continuous improvement and adaptation.

If the architecture development for enterprise integration follows an engineering approach, then the process can also be called enterprise engineering. When the practice of enterprise architecture is enterprise engineering, it can be considered a case of system of systems engineering of the socio-technical system that embodies the enterprise. However, GERAM also intends to be suitable for describing evolutionary scenarios applicable to complex systems that can be the result of emergent rather than managed and engineered change.

GERAM intends to facilitate the unification of methods of several disciplines used in the change process, such as methods of industrial engineering, management science, control engineering, communication and information technology, i.e. to allow their combined use, as opposed to segregated application.

One feature of the GERAM framework is that it unifies the two distinct approaches of enterprise integration, those based on product models and those based on business process design. It also offers new insights into the project, programme, and portfolio management of enterprise integration and the relationship of integration with other strategic activities in an enterprise.

An important characteristic of enterprise architecture is the recognition and identification of feedback loops on various levels of enterprise performance as they relate to its products, mission and strategic goals. To achieve such feedback with respect to both the internal and the external environment, performance indicators and evaluation criteria of the corresponding impact of change are necessary. The continuous use of these feedback loops is a prerequisite for the continuous improvement process of the enterprise operation and its adaptation to existing or anticipated changes in the relevant market, in technology, as well as internal and external social, organizational, economical, and ecological environment.

B.2 Framework for enterprise architecture and enterprise integration

B.2.1 Framework overview

GERAM provides a description of all the elements recommended in enterprise architecting, engineering and integration and thereby establishes a threshold for the collection of tools and methods from

which any enterprise would benefit to more successfully tackle initial integration design and the change processes that can occur during the operational lifetime of the enterprise. It does not impose a specific set of tools or methods, but defines the satisfaction criteria for any set of selected tools and methods. GERAM considers enterprise models as an essential component of enterprise engineering and integration; this includes various formal and less formal forms of design descriptions utilized in the course of design, as described in enterprise engineering methodologies, such as computer models, and text- and graphics-based design representations.

The set of components identified in GERAM is shown in [Figure B.1](#), while [B.2.2](#) defines each component, which is further specified in [B.3](#).

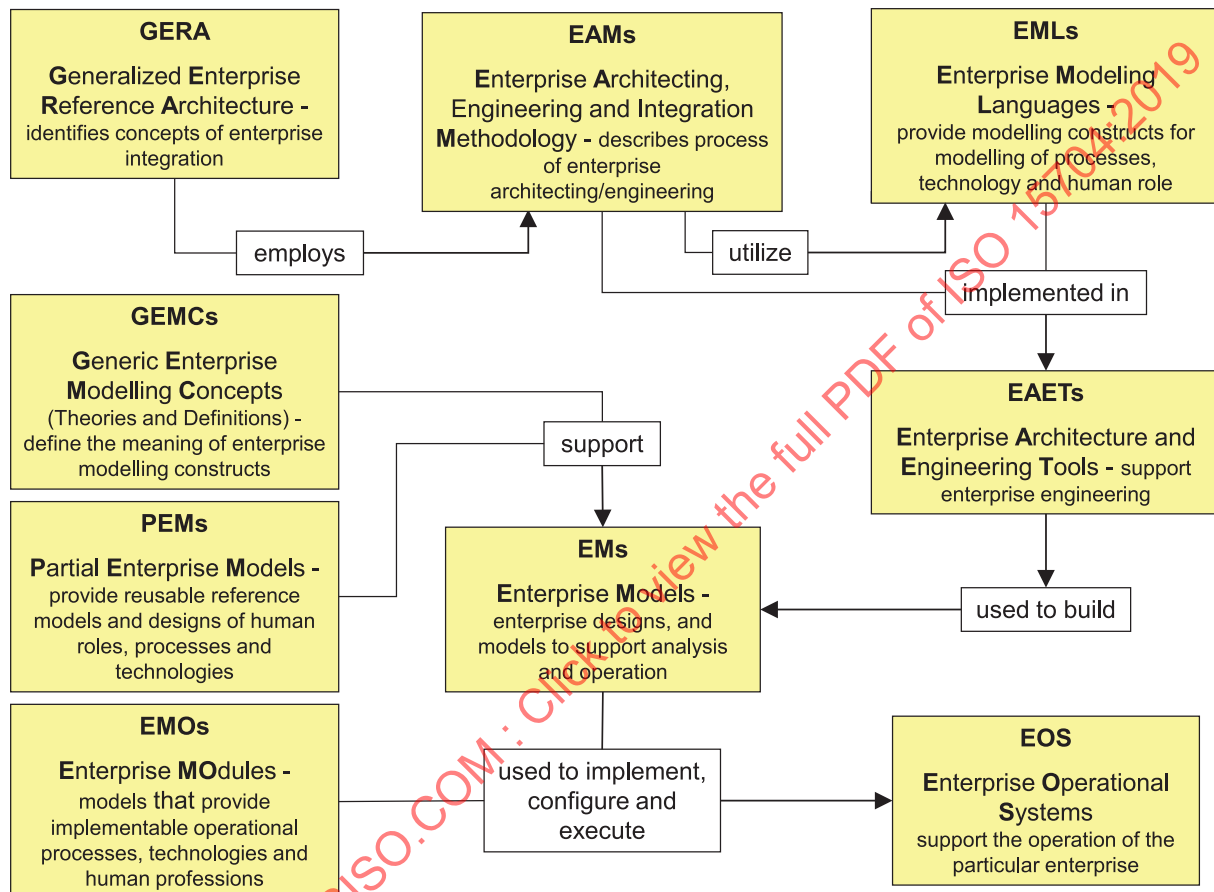


Figure B.1 — GERAM framework components

The GERAM framework identifies in its most important component, the Generalized Enterprise Reference Architecture (GERA), the basic concepts for use in enterprise engineering and integration, e.g. enterprise entities, life cycles and life histories of enterprise entities. GERAM distinguishes between the methodologies for enterprise architecture (EAMs) and the modelling languages (EMLs) used by the methodologies to describe and model the structure, content and behaviour of the enterprise entities. These languages enable the modelling of the human roles in enterprise operation as well as business processes and their supporting technologies. The modelling process produces enterprise models (EMs) that represent all or part of enterprise operations, including its manufacturing or service tasks, its organization and management, and its control and information systems. These models are used to implement, configure and execute the enterprise operational system (EOS) as well as to improve the ability of the enterprise to evaluate operational or organizational alternatives, e.g. by simulation, and thereby enhance its current and future performance.

The methodology and the languages used for enterprise modelling are supported by enterprise architecture and engineering tools (EAETs), enabling enterprise model creation, sharing,

documentation, information management that includes versioning, status, and presentation to various stakeholders, and in general the processes involved in the governance of any architecting effort.

Glossaries, meta-models and ontological theories can define the semantics of the modelling languages and collectively constitute the generic enterprise modelling concepts (GEMCs). Using partial enterprise models (PEMs), which are reusable models of human roles, processes and technologies, enhances the modelling process.

Specific enterprise modules (EMOs) are pre-existing components and products like human skill profiles for specific professions or common business procedures, e.g. banking and tax rules, or IT infrastructure services, or any other product useful for supporting the operational use of enterprise models as components in the implementation of the operational system (EOS).

Potentially, GERAM can characterize all proposed EA Frameworks and therefore developers of specific frameworks can gain from being able to commonly refer to the capabilities of their frameworks without having to rewrite their documents to comply with GERAM. End users benefit from GERAM because the GERAM definitions allow them to identify what they can and cannot expect from a specific architecture framework, e.g. in connection with an enterprise integration methodology and its proposed supporting components.

B.2.2 GERAM framework component definitions

B.2.2.1 Generalized Enterprise Reference Architecture (GERA)

GERA defines a number of enterprise related generic concepts recommended for use in enterprise architecture/engineering and integration endeavours. These concepts can be categorized as:

- human-oriented concepts
 - to describe the human roles as an integral part of the organization and operation of an enterprise, and
 - to support humans during enterprise design, construction, and change;
- process-oriented concepts for the description of the business processes for mission fulfilment as well as management and control of the enterprise;
- technology-oriented concepts for the description of the supporting technology involved in both enterprise operation, e.g. production, management and control, and in enterprise engineering efforts, e.g. modelling and model use support.

B.2.2.2 Enterprise architecture and integration methodologies (EAMs)

EAMs describe the processes of enterprise engineering and integration. A methodology may be expressed in the form of a process model or structured procedure with detailed instructions for each enterprise engineering integration activity and task, as well as characteristics of the desired outcome.

B.2.2.3 Enterprise modelling languages (EMLs)

EMLs define the modelling constructs for enterprise modelling adapted to the needs of people creating and using enterprise models. Specifically, enterprise modelling languages provide constructs to describe and model human roles, operational processes and their functional contents as well as the supporting information, office and production technologies. GERAM expects that selected enterprise modelling languages have the expressive power to ensure the models created can answer relevant questions and concerns of stakeholders. The representations produced using the languages should be uniformly interpreted and understood by all relevant stakeholders.

B.2.2.4 Generic enterprise modelling concepts (GEMCs)

GEMCs define and formalize the most generic concepts of enterprise modelling with definitions occurring in various ways. In increasing order of formality generic enterprise modelling concepts may be defined as:

- a) natural language explanation of the meaning of modelling concepts (glossaries);
- b) some form of meta-model describing the relationship among modelling concepts available in enterprise modelling languages;
- c) ontological theories defining the meaning, i.e. semantics, of enterprise modelling language concepts.

B.2.2.5 Partial enterprise models (PEMs)

PEMs are reusable, paradigmatic, typical models that capture characteristics common to many enterprises within or across one or more industrial sectors. These models capitalize on previous knowledge by allowing the development of model libraries and reuse in a “plug-and-play” manner, rather than developing the models from scratch. Partial models make the modelling process more efficient.

The scope of these models extends to all possible components of the enterprise such as:

- models of human roles, e.g. skills and competencies of humans in enterprise operation and management;
- operational processes, e.g. functionality and behaviour;
- technology components, e.g. service or manufacturing oriented;
- infrastructure components, e.g. information technology, energy, services.

A partial model may cover the whole or a part of a typical enterprise, and may concern various enterprise entities such as products, projects, companies, and may utilize various representations such as data models, process models, organization models, to name a few. Partial models may also exist at various extents of abstraction, describing typical business models, capabilities, and architecture principles. The partial model designation also includes reference models for typical architectural solutions, design patterns, etc.

B.2.2.6 Enterprise architecture and engineering tools (EAETs)

EAETs support the processes of enterprise architecture, engineering and integration by supporting an enterprise engineering methodology and offering enterprise modelling capabilities, e.g. model creation, collaboration, analysis, sharing, and library and repository functions. Engineering tools should provide for analysis, design and use of enterprise models.

B.2.2.7 Enterprise models (EMs)

EMs represent particular enterprise entities. Enterprise modelling languages can best express enterprise models. EMs include various designs, e.g. models prepared for analysis or design, and executable models to support the operation of the enterprise, e.g. model-based control or workflow execution. For a particular enterprise, a set of EMs may consist of several models describing the enterprise from various perspectives and aspects, and at different extents of abstraction.

B.2.2.8 Enterprise modules (EMOs)

EMOs are either products or human roles for utilization as building blocks in the implementation of the enterprise. Examples of enterprise modules intended to support the operational use of enterprise models are human role resources with given skill profiles for specific professions, types of manufacturing and

logistic resources, common business equipment or applications, databases, communication, control, computing (IT) software and hardware/equipment/devices.

B.2.2.9 Enterprise operational system (EOS)

An EOS is an operational implementation of a particular enterprise. A particular enterprise model, which provides the system specifications and identifies the enterprise modules used in the implementation of the particular enterprise system, guides an EOS implementation. An EOS is typically a socio-technical system of systems.

B.3 GERAM framework component specification

B.3.1 GERA specification

B.3.1.1 Fundamental concepts

GERA defines the generic concepts recommended for use in enterprise architecture/engineering and integration projects of all sizes and varieties. GERA classifies these generic concepts, as outlined below, to focus attention on the need for a comprehensive scope in enterprise modelling, and consequently to respond to the widest range of stakeholder perspective concerns. GERA classifies these generic concepts as outlined below.

- a) Process-oriented concepts cover enterprise operations including functionality and information flow among functions over the enterprise life cycle and activities in various life cycle phases, the life history of the enterprise, types of participating enterprise entities, and enterprise modelling with integrated model representation and model views.
- b) Technology-oriented concepts cover various infrastructures used to support processes and include among others the resource models, e.g. information technology, manufacturing technology, and office automation, facility layout models, information system models, communication system models, logistics models and models of different kinds of equipment used in mission fulfilment.
- c) Human-oriented concepts cover human aspects such as adequate capabilities, skills, know-how and competencies as well as roles of humans in the enterprise organization and operation. The organizational relationships have to do with decision level, responsibilities and authorities, while the operational relationships have to do with the capabilities and qualities of human roles as enterprise resource elements. To address interoperation with other humans and with technology elements when realizing enterprise operations, the architecture needs to recognize the communication requirements of humans.

Modelling constructs are necessary to facilitate the description of human roles as an integral part of the organization and operation of an enterprise. The constructs should facilitate the capture of enterprise models that describe

- human roles,
- the way in which human roles are organized so that they interoperate with other human and technology elements when realizing enterprise operations, and
- the capabilities and qualities of human roles as enterprise resource elements.

An appropriate methodology, see [B.3.2](#), is necessary to promote the retention and reuse of models that encapsulate knowledge, i.e. know-how possessed by humans expressed as an enterprise asset, which is an important consideration in enterprise engineering and transformation projects.

Examples of enterprise reference architectures, as architecture frameworks, are provided by Architecture of Integrated Information Systems (ARIS)^[62], CIMOSA^[36], GRAI/GIM^[34], Integrated Enterprise Modelling (IEM)^[49], PERA^[71], and many more that supplement or are derivatives of

them. ISO 19439 defines a general Framework for Enterprise Modelling and this document defines requirements that such frameworks need to satisfy.

B.3.1.2 Human-oriented concepts

B.3.1.2.1 Human roles

The role of humans in the enterprise remains fundamental and foundational. However sophisticated and integrated an enterprise can be, humans will always make the final decisions. Enterprise is a particularly human phenomenon. With the emergence of decentralized organizational structures, flat hierarchies and responsibility and authority delegation, the knowledge about the roles of individuals and who is responsible for what becomes an invaluable asset for any enterprise, especially those operating according to new management paradigms. Therefore, capturing this knowledge in enterprise models proves to be very useful and enables flexible adaptation to existing or expected environmental changes. In addition, capturing the different factors describing the capabilities for human roles is necessary. Human factors include professional skills, experience, social and cultural elements, etc.

Typically, humans may assume different roles during enterprise engineering and operation. Examples are: chief executive, chairperson, marketing, sales, technical researchers and developers, finance, engineering and manufacturing directors, product design, production planning, information systems, quality, product support, logistics, capital equipment, shop floor and site managers, assistant managers, accountants, cashiers, product, process and information system designers, production engineers, electrical and mechanical technicians, maintenance personnel, quality inspectors, supervisors and foremen, machine operators, storeroom and inventory persons, progress chasers, secretaries, drivers, cleaners, management and systems consultants, systems integrators, system builders, and IT suppliers and vendors.

Often humans and groupings of humans are given a number of roles and responsibilities to carry out concurrently and cohesively, where each can involve different reporting lines and control procedures. Furthermore, their roles can be expected to change over time as process requirements change and individual and group capabilities advance or decline. The ability to manage and deploy human role resources effectively and collectively under complicated and changing circumstances is key to the competitive position of an enterprise.

While modelling all characteristics of human roles within an enterprise is not practical, concepts are necessary to formally represent those human factors connected with enterprise integration. This representation should achieve the harmonization of human roles with that of other human roles and technology elements, as an integral part of the organization and operation of an enterprise. Hence the need for constructs that promote the capture of knowledge possessed by humans in the form of reusable enterprise models about:

- a) the role of individuals and groups of individuals;
- b) the way in which organizational structures and constraints coordinate those roles, e.g. delegation of responsibilities and control and reporting procedures;
- c) the role capabilities and qualities of humans, treated like resource elements;
- d) the development of individuals, their skills, knowledge and career paths.

Importance should be given to understanding when, by whom and how decisions are made in the enterprise as well as who can fulfil certain tasks in the replacement of others. Therefore, current human role models, including those related to organizational learning, have important use during the operation of the enterprise, not only for enterprise transformation activities.

Knowledge about the roles of humans and ways in which those roles can be harmonized should be capitalized and reused as an enterprise asset. The extent to which such knowledge becomes formalized within computer processable models directly influences the extent to which knowledge can be capitalized. Computer processable models based on common understanding naturally facilitate analysis, transformation, storage and integration, whereas mental models retained and processed by

humans are less tractable for such purposes. However, the retention and reuse of informal models, e.g. cause and effect relationships and shared mental models or images, can provide benefit in realizing improved enterprise cohesion. Hence even where formal modelling of human issues proves impractical, deploying suitable social processes, human role organizational structures, methodologies and tools that promote explicit model capture and visualization should encourage the retention and reuse of knowledge. Even when explicit models are not available or able to express the knowledge itself, expressing the characteristics of the competency relying on that knowledge supports strategic human role resource decisions.

The ability to retain and reuse human factors knowledge can be of vital importance to the competitive position of an enterprise. Its reuse can enable an enterprise to:

- respond rapidly to new market opportunities or changes in environmental conditions;
- reengineer its business and manufacturing processes;
- improve its management and utilization of resources as new products and services are launched;
- improve its resilience to the loss of core competencies that substantially rely on knowledgeable human assets.

B.3.1.2.2 Models of human roles

A taxonomy of human factors and their relation to activities and processes in which humans participate would allow human aspects to relate to enterprise models. Human role models are necessary for decision making, capability and skill models, organizational models, socio-technical models, e.g. motivation and incentives, and others when available.

Human role models support the definition of human responsibilities and authorities in both the enterprise operation and its organizational structure. Such models support the collection of relevant role information and the recognition of roles in the design of the operational system. The viewpoint concept accommodates human factors in GERA. This concept provides in its process-oriented model views and technology-oriented implementation view for the recognition of human roles and the capturing of relevant role information. These views also recognize the human roles as an operational resource with descriptions of human skills and capabilities. Enterprise integration also needs to deal with the accommodation of human characteristics in the change methodology, both in the human role of change agent and in the role of potential and actual resource.

Advances in automation systems technology offer the opportunity for automata to augment or supplant human roles. Therefore, human role models and models of automata roles become important considerations for trade-off analysis (see [B.3.2.2](#)).

B.3.1.3 Process-oriented concepts

B.3.1.3.1 Process modelling concepts

Business process-oriented modelling aims to describe the processes in the enterprise by capturing both their functionality, i.e. what tasks to perform, and their behaviour and all other characteristics relevant in the context (throughput, speed, reliability, precision, various statistical properties, etc.), including when and in which sequence to perform those tasks. To achieve a sufficiently complete description of the processes, the guiding methodology (see [B.3.2](#)) recognizes a number of essential concepts. GERA identifies the following process-oriented concepts:

- a) enterprise life cycle and life cycle phases, which describe the kinds of activities involved in life cycle processes of enterprise entities;
- b) life history, which describes the instantiation in time of life cycle activities;
- c) kinds of entities within the enterprise and their roles in each other's life cycle processes;

- d) enterprise modelling with integrated model representation using modelling viewpoints that can guide modelling and generate relevant model views.

The following sections describe these concepts in more detail.

B.3.1.3.2 Life cycle concept

B.3.1.3.2.1 Life cycle phases

Figure B.2 shows the GERA life cycle for any enterprise or any of its entities. The different life cycle phases define collections of related activities that are pertinent during the life of the entity. Life cycle activities encompass all activities from identification to decommissioning at end of life of the enterprise or entity. GERA defines seven life cycle activity collections characterized by dependency relationships rather than chronology. A phase collection can subdivide further as demonstrated for the design activities that partition into two sub-collections of activities based on the customary subdivision in many industries of design into preliminary and detailed design activities. The life cycle diagram depicting the life cycle of an entity is itself a highly abstracted model of the enterprise engineering methodology.

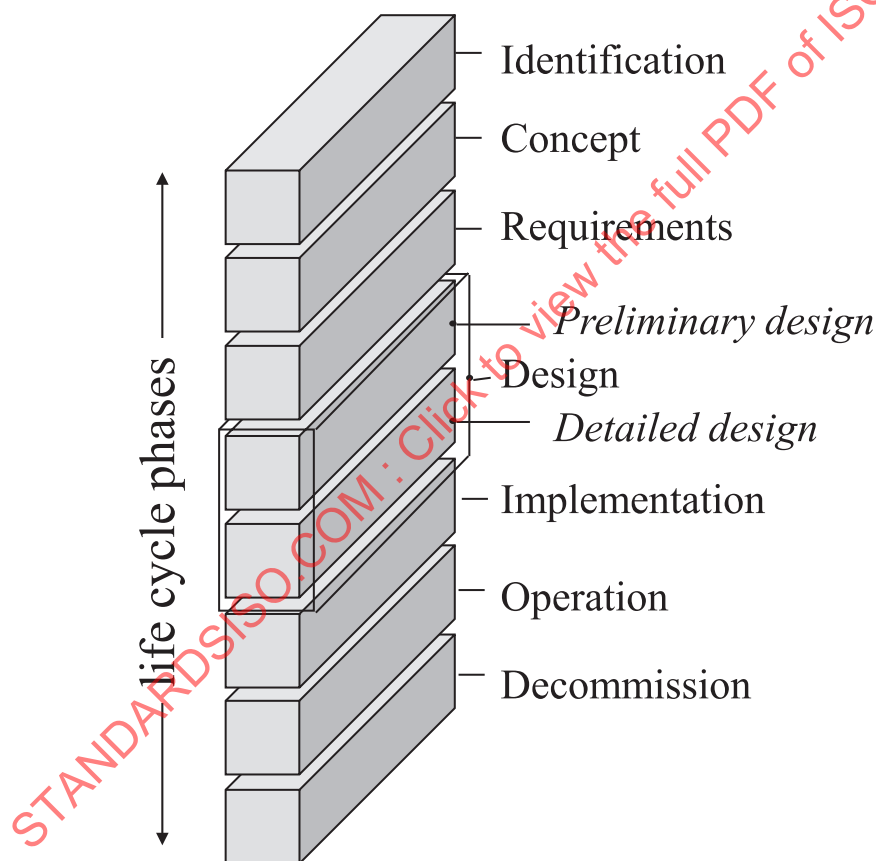


Figure B.2 — GERA life cycle phases

B.3.1.3.2.2 Entity identification

Entity identification consists of the set of activities that identify the context of the entity under consideration in terms of its boundaries and its relation to its environment. These activities include the identification of the existence and nature of a need, or the need for change of identity. Usually upper management performs these activities because the identity of the entity has strategic significance.

B.3.1.3.2.3 Entity conceptualization

Entity conceptualization consists of the set of activities necessary to develop the business concept of the underlying entity, including the definition of the entity's vision, mission, values, strategies and strategic relationships, objectives, capabilities, operational concepts, policies, fundamental principles and business plans. Usually upper management performs these activities with support from a range of strategic analyses.

B.3.1.3.2.4 Entity requirement definition

Entity requirement definition consists of the set of activities necessary for developing descriptions of operational requirements of the instantiated enterprise, its relevant processes and the collection of all their functional, behavioural, informational and capability needs. This description includes both service and manufacturing/mission fulfilment requirements and management and control requirements of the entity, without consideration of provisioning by humans as individuals or organizational entities, or by machinery as manufacturing, information, control, communication, or any other technology. Normally, management specifies overarching requirements, often expressed as the mandate of the entity presented as definitions for appropriately characterized high level entity functions, and business analyst develop detail requirement specifications.

B.3.1.3.2.5 Entity design

Entity design consists of the activities necessary for supporting the specification of the entity with all its components, which satisfy the enterprise requirements. The extent of design activities includes the design of all human role tasks, those of individuals and of organizational entities, and all machine tasks concerned with the entity's mission fulfilment, i.e. service to customers or production, and the related management and control functions. The design of the operational processes includes the identification of the necessary resources, which includes manufacturing, information, communication, control or any other applicable technology.

Any life cycle phase can be subdivided to provide additional structuring of life cycle activities such as:

- a) preliminary design of the overall enterprise specification that is sufficient to establish the feasibility, cost, time, and risk for securing management approval to proceed with the creation of entity transformation;
- b) detail design work necessary for the complete system design specification suitable for building the final physical or virtual system that includes verification mechanisms.

Overall enterprise specification is a conceptual preliminary design activity that is fundamental to the development of the solution to the problem presented by the set of requirements. The term solution architect is often used for those performing this activity. From an upper management perspective, architecture development is an activity. However, from the solution architect's perspective, the details of this activity involve several technical, governance, and management processes, see ISO/IEC/IEEE 42020 and ISO/IEC/IEEE 42030.

B.3.1.3.2.6 Entity implementation

Entity implementation consists of the set of activities that define all those tasks that are necessary to build or re-build, i.e. manifest, the entity. This comprises implementation in the broadest sense, which covers:

- a) commissioning, purchasing, (re)configuring and deploying all service, manufacturing and control software as well as hardware resources;
- b) hiring and training personnel, and developing or changing the human organization;
- c) construction, manufacturing, or building the entity;
- d) component testing and verification, system integration, testing and validation;

- e) releasing the newly deployed entity or its new version into operation;
- f) creation of or update to the as-built documentation or implementation description.

NOTE 1 The implementation description or documentation can deviate from the design specification of the entity due to preferences or unavailability of specified components.

NOTE 2 Conceptually, component and integration testing is not necessarily considered part of the implementation activity, because for testing it is necessary to create a build and release it into a test-operational environment, and then operate the component, or the entire system, for testing purposes. In such cases, implementation is subdivided into separate build and test collections of activities.

B.3.1.3.2.7 Entity operation

Entity operation consists of the set of activities necessary for producing the customer's product or service, which is the entity's special mission along with all those tasks necessary for monitoring, controlling, and evaluating the operation. Deviations from goals and objectives or any feedback from the environment can lead to the identification of the need for change, which may include enterprise re-engineering or continuous improvement of its human role and technology resources, its business processes, and its organization.

NOTE The actual change activities consist of instances of life cycle activities [B.3.1.3.2.2](#) through [B.3.1.3.2.6](#). In a similar manner, operational needs often require dynamic configuration or reconfiguration of parts of the system, or alternatively, the creation of a virtual organization.

B.3.1.3.2.8 Entity decommissioning

Entity decommissioning consists of the set of activities necessary for decommissioning, recycling, preservation, transfer, disbanding, disassembly, or disposal of all or part of the entity. Decommissioning activities occur in small and large upgrades to the EOS, as well as when the EOS is completely disbanded at the end of its useful life in operation.

NOTE Redeployment is a sequence of activities that involves decommissioning of a resource from a role, possibly followed by (re)training, and subsequent deployment in a different role.

B.3.1.3.3 Life history

The life history of an enterprise entity is a representation of the chronological sequence of activities experienced by a particular entity during its entire lifetime. Relating to the life cycle concept described above, the concept of life history allows identification of the tasks pertaining to these different phases of activity. This relationship demonstrates the iterative nature of the life cycle concept compared with the time sequence of life history.

The instance of a typical change process consists of a coordinated set of life cycle activity instances (also referred to as a sequence of events) and interspersed with milestones that sometimes act as gates to subsequent activities.

Moreover, change processes can interact with one another. Within one process, such as a continuous improvement project, multiple life cycle activities usually are active at any one time, e.g. concurrent engineering design and implementation processes can execute within one enterprise engineering process with considerable time overlap, and typically in parallel with the enterprise operation.

Life histories of entities are all unique, but all histories record processes that in turn rely on the same kind of life cycle activities as defined in the GERA life cycles. For this reason, life cycle activities are a useful abstraction in understanding the life history of any entity.

EXAMPLE [Figure B.3](#) illustrates the relations between life cycle and life history representing a simple case with a total of seven processes: three engineering processes, three operational processes, two partial decommissioning processes and one decommissioning process.

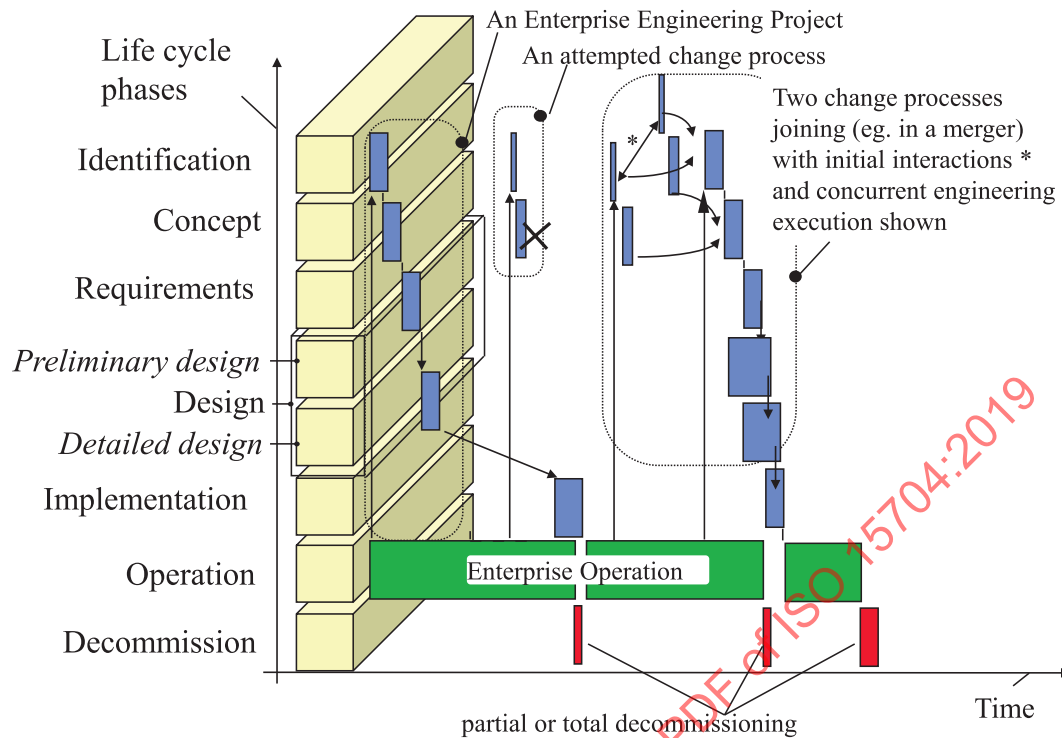


Figure B.3 — Iterative processes in entity life-history

B.3.1.3.4 Entity interaction in enterprise integration

Figure B.4 shows one of several ways in which the life cycle activities of two entities can interact. The operation of Engineering Project entity supports the life cycle activities for design and implementation of the Factory entity, e.g. the Engineering Project entity may be an engineering project that designs the factory for the Factory entity.

Conversely the life cycle activities of the Engineering Project entity need information about the life cycle details of the Factory entity, i.e. another entity, such as a factory owner, needs to identify a plant, define its concepts and requirements, such as information about which life cycle activities of the plant's products are to be performed by the operation of this plant.

Many other relations between the life cycle activities of enterprise entities are possible. However, in every case only the operational activities of entities influence the life cycle activities of other entities. GERA introduces the concept of entity type and the relations between the different entity types. While many categories of enterprise entities are possible to define as entity types, the following discusses two different ways of categorizing enterprise types:

- a) an operation-oriented set;
- b) a generic and recursive set.

The two sets have close relations to each other and both identify the product entity as the result of the operation of other entities.

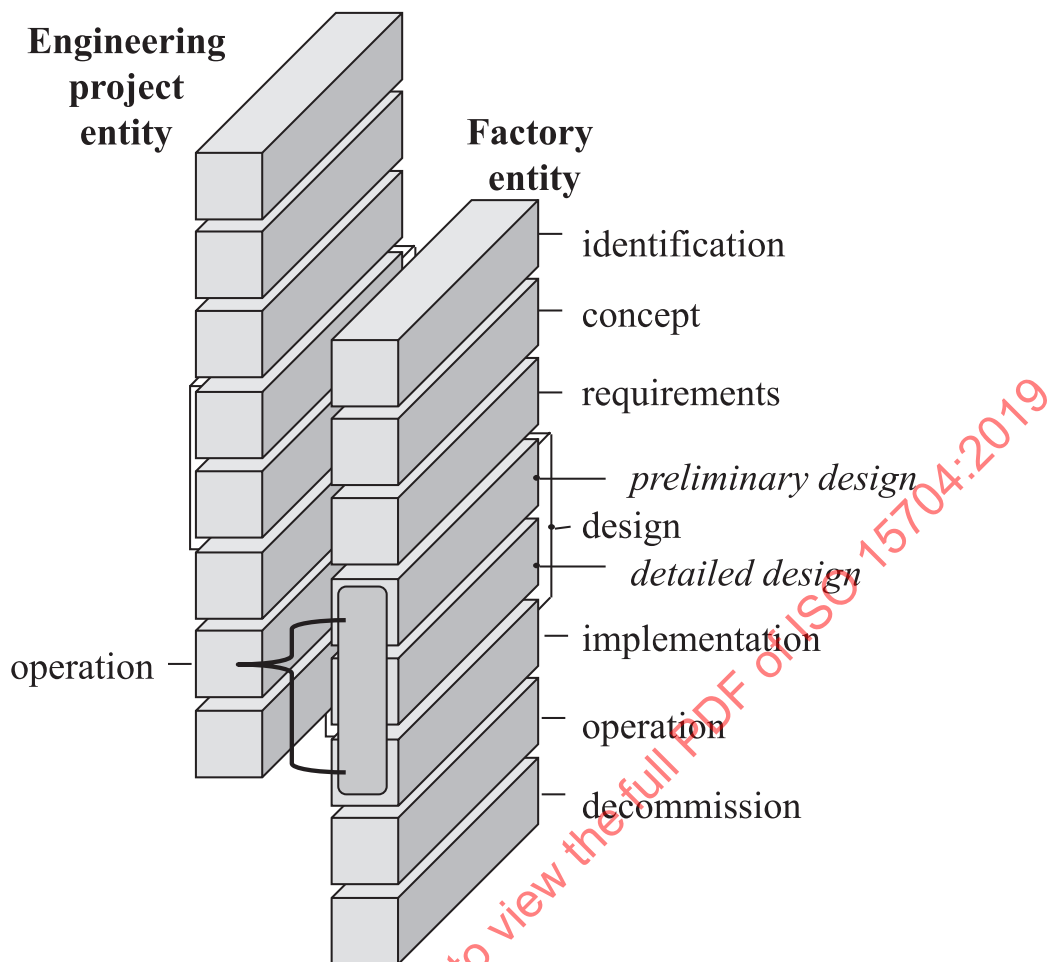


Figure B.4 — Engineering Project and Factory life cycle interaction

B.3.1.3.5 Project-oriented enterprise

The project-oriented enterprise, often with a short life history, meets the need for the one-off production of another entity, e.g. enterprise engineering project, one of a kind manufacturing project, EPC (engineering, procurement, construction) project, building projects.

Close linkage with the life cycle of the single product or service that it is producing or changing characterize the project enterprise. The management system of project enterprises is typically set up quickly, while the rest is created and operated in stages in lock-step with the life cycle activities of the product of the project.

Project enterprises are normally associated with, or created by repetitive service and manufacturing enterprises, e.g. an engineering project created by an engineering enterprise.

The products of project enterprises may be diverse, such as large equipment, buildings etc., or an enterprise (e.g. a plant, or an infrastructure enterprise).

NOTE Many very large infrastructure projects become a set of hierarchical sub-projects devoted to specific aspects of the larger project, e.g. a new long distance high-speed highway requiring commissioning, preliminary design, land acquisition, detail design, drainage, pavement, bridges, etc. These large projects are thus a system of systems.

B.3.1.3.6 Repetitive service and manufacturing-oriented enterprise

The repetitive service and manufacturing-oriented enterprise typically supports a family of products, produced in a repetitive or sustained mode. During their life history these business enterprises undergo multiple change processes, e.g. service enterprises, manufacturing plants, engineering firms, infrastructure enterprises.

The products of the repetitive service and manufacturing-oriented enterprise may be diverse, such as non-enterprise product entities (see [B.3.1.3.7](#)); or products that are enterprises themselves, e.g. project enterprises are regularly created by engineering and building companies.

B.3.1.3.7 Product entity

The product entity refers to a very large class of entities including any artificial product, such as customer goods, services, hardware equipment, computer software, etc. These entities are not enterprises themselves, but GERAM describes their life cycles and their life histories can be arranged into the life cycle stages, as identified in ISO/IEC/IEEE 15288.

NOTE Technically speaking, a service product is not a product entity, but is a service process arranging a set of service functions performed by a service entity (see [B.3.1.3.8](#)) in support of a process of the service user entity. To achieve this support for the user process, the service entity executes a service process that utilizes the service entity's resources to implement the desired function.

B.3.1.3.8 Enterprise recursion

This sub-clause describes a generic and recursive set of typical enterprise entity types in the sense that the first entity generates, creates or changes the second, the second generates, creates or changes the third, etc., and each generation has the same basic structural form. The following provides typical recursive relationships of this kind.

- a) Strategic Management entity: defines the necessity and the starting of any enterprise engineering/integration effort.
- b) Engineering/Implementation entity, e.g. project or programme: provides the means to carry out the enterprise engineering efforts. The project or programme may use the services of a Methodology entity like a consulting service, to define, design, implement and build the operation of the enterprise entity, which may be a new production line, or a single product like a ship.
- c) Production entity, e.g. factory or shipyard: is the result of the operation of the Engineering/Implementation entity. The Production entity uses the operational system created by the Engineering/Implementation Entity to design and build the products, e.g. a ship, and/or to provide the customer services of the enterprise.
- d) Product entity: is the result of the operation of Production entity. The Product entity represents the products of the enterprise.

As a complement to the above entity types, a fifth and possibly sixth entity type may provide the methodology needed for guiding the enterprise engineering, enterprise integration, and enterprise operation activities.

- e) Methodology entity: provides the methodology for use by other types of enterprise during their operation, which operation in general leads to the creation of another entity type. Namely, both the strategic management entity and the engineering project may use such methodological support. Management and engineering consulting companies are typically this kind of enterprise.
- f) Engineering and Construction Contractors: represented as one for simplicity, contribute to the technical work of Engineering/ Construction projects or programmes.

[Figure B.5](#) shows the most important generative relationships among the involved entities, plus some operational interactions. These generative relationships expose the fact that one entity's operation contributes to one or more of the non-operational life cycle activities of another entity, thereby

supporting the creation or change of that entity. Typically, additional operational interactions exist, through both the Strategic Management entity and the Production entity contributing to the governance of projects and programmes. If the project's mandate is factory maintenance or factory modernization, then some additional or partial decommissioning will also occur.

Identifying the role of the different entities, their products and the relations between them demonstrates the recursive characteristic of the first four entity types: a) to d). [Figure B.5](#) depicts the recursive chain of enterprise entity development.

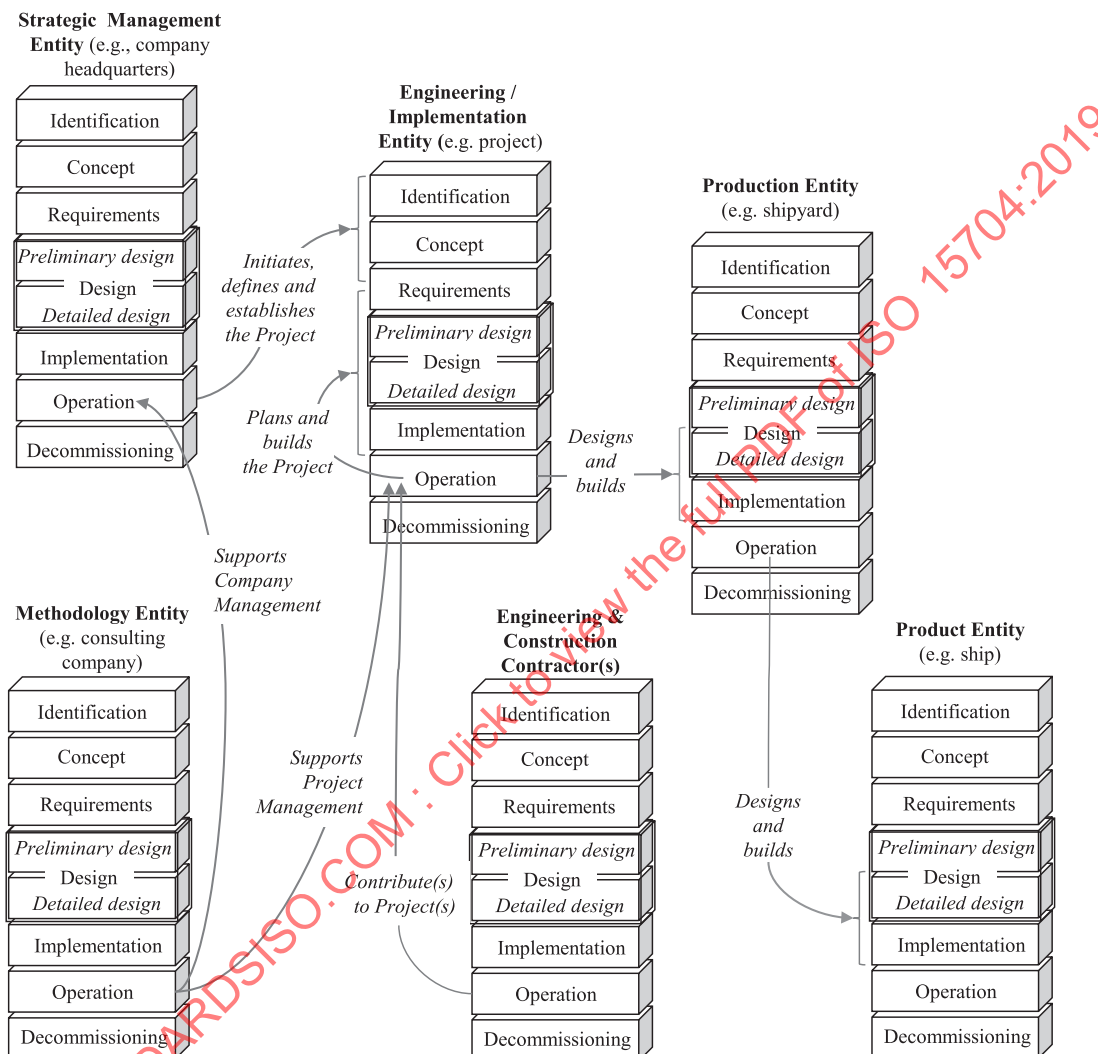


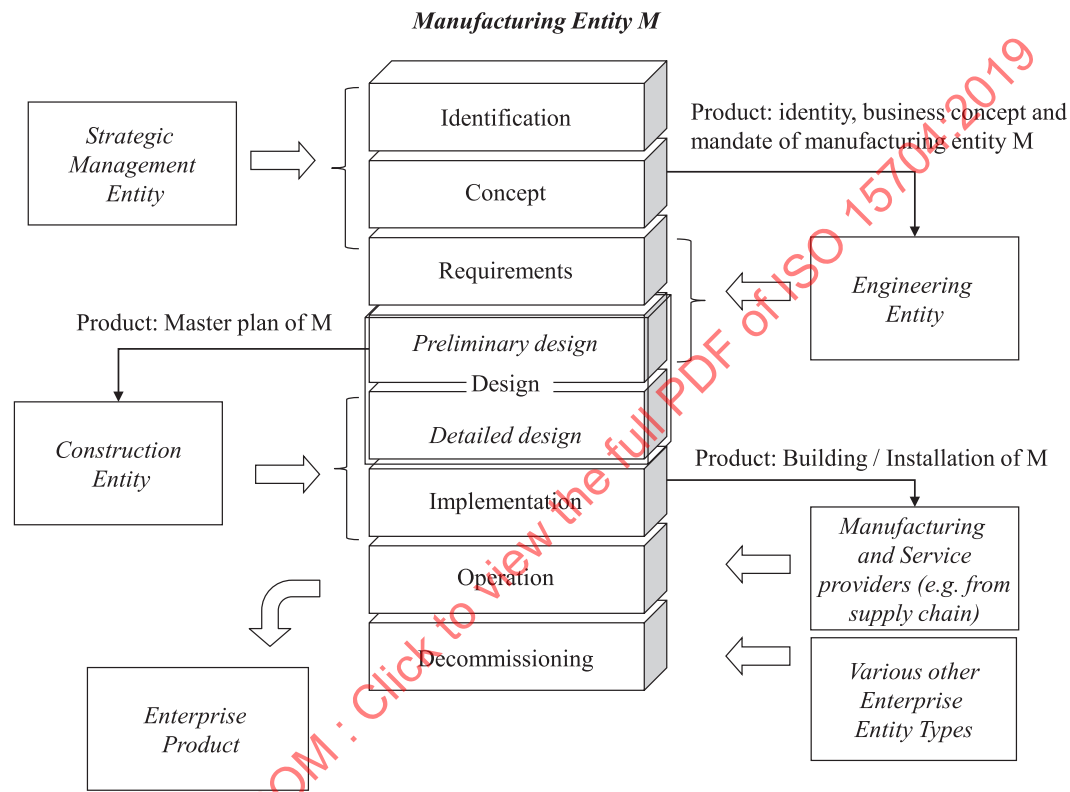
Figure B.5 — Recursive classification of Enterprise entity

The Strategic Management entity will always start creation of any subordinate entity by identifying scope, goal and objectives for the specific entity. Development and implementation of a new enterprise, or new business unit, is then done by an Engineering/Implementation entity whereas a Production entity is responsible for developing and manufacturing a new Product entity. All enterprise entities have an associated and clearly identifiable entity life cycle. The operational part of the entity life cycle always defines, creates, develops and builds the successor entity. The operation itself may be supported by an associated methodology for enterprise engineering, enterprise operation, product development and production support. These methodologies are usually expressed in some form of process definition and used as a reference model, but possibly extended with models for other aspects, e.g. information models, organizational models.

Figure B.5 illustrates that the methodology entity also has a life cycle. There is a clear distinction between

- the life cycle of the methodology entity, which is essentially the description of how a methodology is developed and embedded as the service delivery process of a consulting company, and
- the manifestation of the methodology during the operation of the methodology entity, in support of the consulting company's clients.

EXAMPLE The generative relations of the different entity types are also shown in Figure B.6, which demonstrates the contributions of different entities to the life cycle of a Production entity. The Production entity itself produces the enterprise's Product entity during its operation phase.



Note: Wide arrows represent life-cycle relationships and narrow arrows represent information products

Figure B.6 — Contributions of entity types to Production entity

B.3.1.3.9 Process modelling

Process modelling is the activity that results in various models of the management and control as well as the service and production processes, and their relationships to the resources, organization, products, etc. of the enterprise. Process modelling allows representation of the operation of entities within the enterprise in all their aspects: functional, behaviour, information, resources and organization, etc. This representation provides for operational use of the models in support of decision-making by evaluating operational alternatives, and for model driven operation control and monitoring.

B.3.1.4 Technology-oriented concepts

B.3.1.4.1 Importance of technology to enterprise

Both the enterprise engineering process and the operational environment employ and are becoming increasingly dependent upon a significant amount of technology. Technology is either production oriented and therefore involved in producing the enterprise products and customer services, or management and control oriented — providing the necessary means for communication, information

processing, decision support and information sharing. Technology-oriented concepts provide descriptions of the technology involved in both the enterprise operation and the enterprise engineering efforts.

For the operation-oriented technology, such concepts need to relate to resource models and resource organization models, e.g. shop floor models, system architectures, information models, infrastructure models, and to communication models, e.g. network models, etc.

All these descriptions are applicable in the enterprise engineering environment as well. In addition, there are specific needs for information technology to support enterprise engineering, e.g. engineering tools, model development services and model enactment services for animation, simulation, and model-based operation control and monitoring.

B.3.1.4.2 Information technology support for enterprise architecture and integration

Information technology support for enterprise engineering architecture and integration as well as enterprise operation should provide three main functions:

- a) model portability and interoperability by providing an integrating infrastructure across heterogeneous enterprise environments;
- b) model driven operational support, e.g. decision support and operation monitoring and control, by providing real time access to the enterprise environment;
- c) model reusability by providing a repertoire of reusable constructs and model elements.

These functions result from the fact that in a system of systems context, like a typical enterprise, creating the necessary models all at once is not possible, e.g. when designing and configuring a system of systems, integrating a large number of pre-existing models is normally necessary.

To enable an integrated real time support of the operation, both the process descriptions and the actual information need to be available in real time for decision support, operation monitoring and control, and model maintenance.

Desirable interoperability needs to be understood, determined in the ecosystem in which strategic management intends to operate, and, as part of strategic management decisions, initiated as preparedness building projects for programmes necessary to achieve this intended interoperability. Pre-existing international or industry information and process standards rarely satisfy all interoperability requirements.

B.3.1.4.3 Enterprise model execution and integration service (EMEIS)

To illustrate the potential use of computer executable models for on-line operation of the enterprise, [Figure B.7](#) illustrates the concept of an integrating infrastructure linking the enterprise model to the real-world systems. Integrating services act as a harmonizing platform across the heterogeneous system environments (IT and others) and provide the necessary execution support for the model. The process dynamics captured in the enterprise model act as the control flow for model enactment. Therefore, access to information and its transfer to and from the location of use is controlled by the model and supported by the integrating infrastructure. The harmonizing characteristics of the integrating infrastructure enables transfer of information across and beyond the organization. Through the semantic unification of the modelling framework, interoperability of enterprise models is assured as well.

NOTE EMEIS has several terms applied in practice, e.g. digital twin, digital thread, and is becoming common with model-based operation for many manufactures of large system, e.g. oil & gas refinery and aircraft, where maintenance services are a critical part of the manufacturing enterprise or frequent configuration changes are necessary.

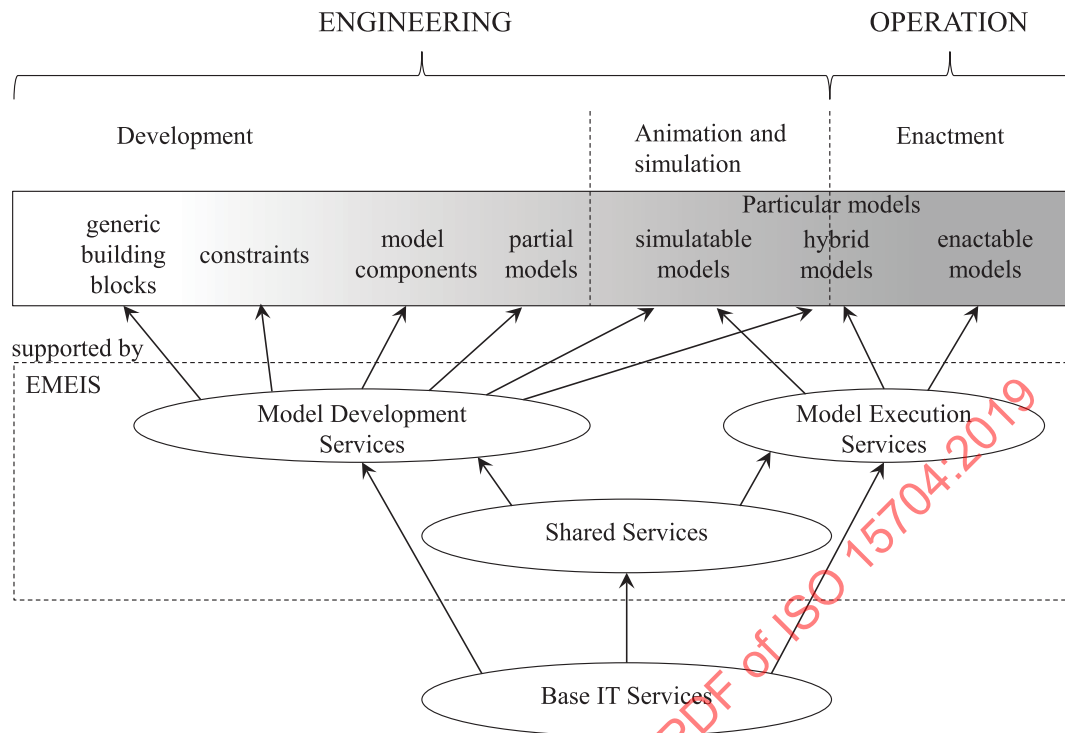


Figure B.7 — Reference model of EMEIS

B.3.1.5 Modelling framework for GERA

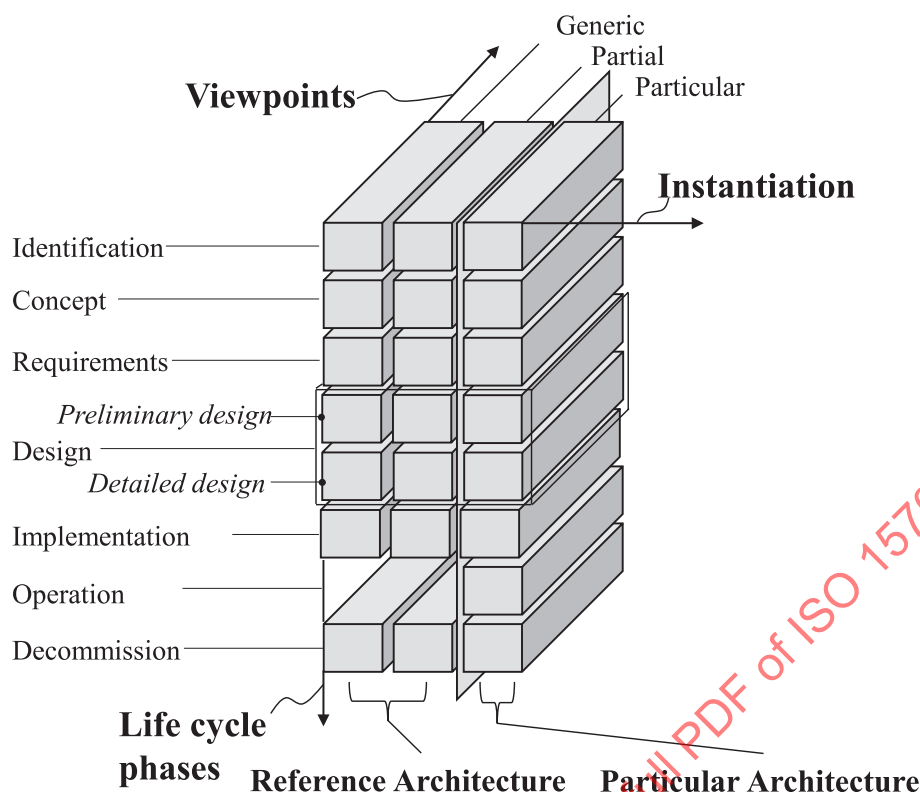
B.3.1.5.1 Framework dimensions

GERA provides an analysis and modelling framework that relies upon the life cycle concept and identifies the following three dimensions for defining the scope and content of enterprise modelling:

- Life cycle dimension — providing for the controlled modelling process of enterprise entities according to life cycle activities;
- Genericity dimension — providing for the controlled particularization (instantiation) process from generic and partial to particular; and
- Viewpoint dimension — providing for the controlled manifestation of specific views of the enterprise.

Figure B.8 shows the three-dimensional structure identified above that represents this modelling framework. The reference part of the modelling framework consists of the generic and the partial extent of genericity only. These two columns structure the definitions of concepts, basic and macro constructs, i.e. the modelling languages, and the model prototypes for description of an area of concern. The extent of genericity represents the results of the modelling process, which is the model or description of the enterprise entity at the state of the modelling process corresponding to the set of life cycle activities.

Throughout the life cycle, modelling languages should support the two-way dependency relationship between models of adjacent life cycle phases. That is, the derivation of models from one state of elaboration to a more elaborated state or the abstraction of elaborated models to one or more model in a less elaborated state can occur without having to create different models for the different sets of life cycle activities. This support capability aids in achieving coherence of models throughout the life cycle.



NOTE The two left side columns represent the reference models; the right side represents the particular enterprise models.

Figure B.8 — GERA Modelling Framework

B.3.1.5.2 Enterprise modelling

Enterprise modelling is the activity that results in partial or particular enterprise models, e.g. various models of the management and control as well as service and production processes, resources, organization, products etc. of the enterprise. The life cycle activities of an entity define one or more models of that entity. The results of enterprise modelling are all the various designs, models prepared for analysis, executable models to support enterprise operation, and so on produced during the lifetime of the enterprise^[27]. The emphasis in enterprise modelling is often on process and product models for representing enterprise operations.

However, there is no preferred or generally applicable kind of process model and associated process modelling language; the selection of the appropriate kind of model depends upon the questions that the model needs to be able to answer and the intrinsic characteristics of the modelled process.

Enterprise models in general represent a very complex reality. In order to reduce this complexity, enterprise models need to enable the representation of certain portions of model content as views of the model's content that are relevant to one or more concerns of the model user. These views allow manipulation of the model for certain aspects of relevance to the user's concerns without being overwhelmed by the overall complexity of the entire model.

Enterprise modelling is not limited to process modelling of the enterprise. All other customary design and analysis activities that create descriptions, or models of the enterprise in any phase of the life cycle also belong to this category, e.g. engineering drawings, charts, and simulations. Information modelling often proceeds or is concurrent with process modelling.

B.3.1.5.3 Model view concepts

B.3.1.5.3.1 Viewpoint concept

To decrease the apparent complexity of the resulting enterprise models, GERA provides the model view concept, which allows description of the operational processes as an integrated model, but with presentation to the user in different sub-sets or model views of an integrated model (see [Figure B.9](#)). Views contain a subset of facts present in the integrated model allowing the user to concentrate on relevant questions that the respective stakeholders can consider using enterprise modelling. Different views are made available highlighting certain aspects of the model and hiding all others. The concept of model view is applicable for models of all entity types across their entire life cycle.

Since the utility of a view depends upon the availability of relevant model content, the concerns of stakeholders are gathered into coherent viewpoints that govern the kind of model appropriate for capturing the necessary content for composing the associated view or views. A viewpoint is the synthesis of a collection of stakeholder concerns into appropriate model view specifications, the creation of which necessitates certain constraints for selecting the appropriate kind of model representation and thus methods for representation of facts about the entity.

The model view concept of GERA assumes an underlying integrated model, which may consist of different kinds of models or partial models of the same kind integrated into a conceptual whole, from which to project appropriate content to satisfy a view specification. Any model manipulation, i.e. any change of the contents of a specific view, are reflected in all relevant views and aspects of the model.

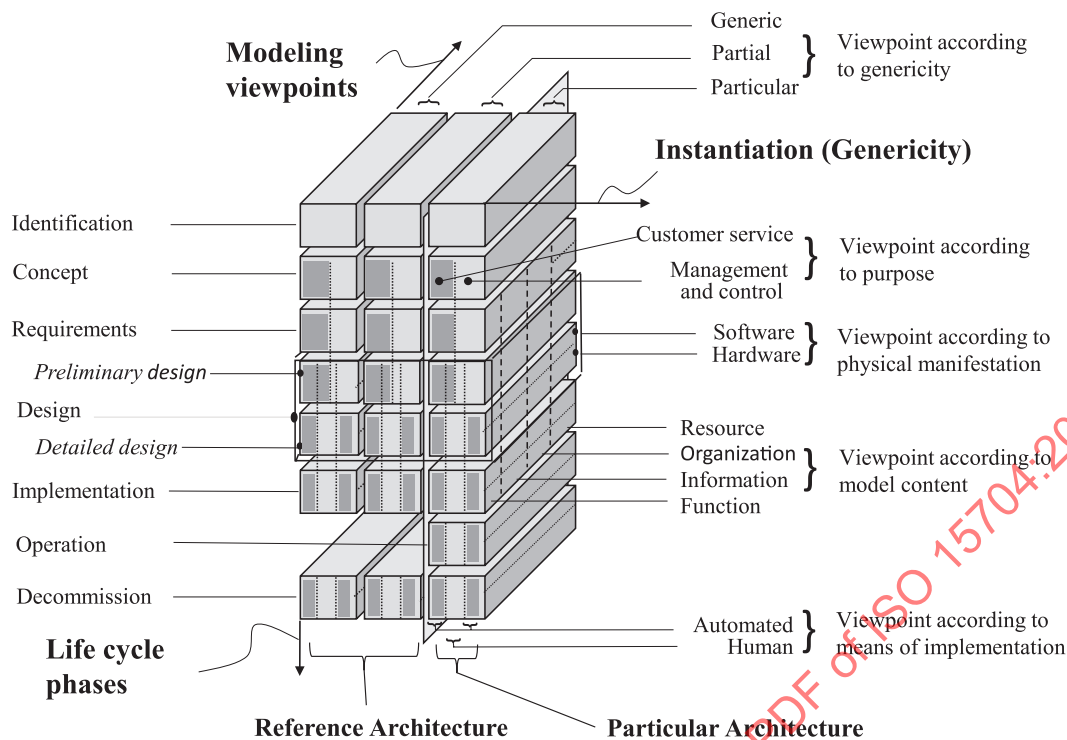
GERA defines a viewpoint for the kinds of models often identified as desirable, allowing for the fact that a GERAM-compliant candidate architecture framework or reference architecture may prescribe even finer grained viewpoints. These viewpoints enable examination of various enterprise aspects from one or more model view. GERA identifies the following viewpoints that represent the different enterprise aspects relative to framework content:

- a) entity Model Contents viewpoint: function, information, resource, organization (FIRO);
- b) entity Purpose viewpoint: customer service and product, management and control;
- c) entity Implementation viewpoint: human implemented tasks, automated tasks for mission support technology, and management and control technology tasks;
- d) entity Physical Manifestation viewpoint: software, hardware.

These aspect-oriented viewpoints identify a few of the many possible ways in which GERA adapts to user and other stakeholder needs and concerns. Additional aspect-oriented viewpoints can be necessary to achieve the purpose of an architecture effort.

GERAM does not require every view to be present in every life cycle phase (see [Figure B.9](#)). However, to assure the capture of all relevant facts, the scope of each aspect defined viewpoint cover other views associated with a viewpoint. For example, it is not as important to have a separate software view and separate hardware view as it is to model both software and hardware. The Enterprise Engineering Methodology decides which model to produce based upon the specifics of a viewpoint collection of stakeholder concerns and which modelling language or formalism to use to describe that model. In other words, the enterprise engineering process needs models for some pragmatic purpose. For example, models can be used to:

- express a design choice;
- simulate a process to find out some process characteristics, such as cost or duration;
- analyse an existing process for finding inconsistencies or other problems in the information or material flow;
- analyse decision functions and find missing decisional roles.



NOTE This figure shows four aspect-oriented viewpoints and associated view contents. Engineering tools can define and support other modelling views.

Figure B.9 — The modelling view concept

The model view concept is the generalization of the view concepts of many architectures beginning with the early efforts of CIMOSA^[36], GRAI^[34], and others and is consistent with the extraction of relevant content from information systems available in current enterprise modelling products and standards, e.g. ISO/IEC 10746, ISO/IEC/IEEE 42010. The GERA modelling framework allows for languages of different expressive power to create model views. This flexibility enables a choice of language in any specific view depending on which analysis capability, and therefore expressive power, is necessary, according to the enterprise engineering methodology's needs.

B.3.1.5.3.2 Entity model content viewpoint

GERA identifies four different model content views for the user-oriented representation of an enterprise entity: Function, Information, Resource and Organization (FIRO).

The Function view represents the functionalities as activities and the behaviour as flow of control of the business processes of the enterprise. This view represents decisional activities of management related operations as well as transformational and support activities. The functional view of the management and control system of an enterprise or entity is indeed the functional model of its decision system.

NOTE The management and control system of the enterprise is often called the decision system.

The function view includes functional models, process models, decisional models, which differ in their expressive power and competency, i.e. in terms of what analysis questions these models can answer, but all address some aspect of enterprise function.

The Information view collects the knowledge about enterprise material and information objects as they are used and produced in the course of enterprise operations. An enterprise information model structures the relationships among objects identified from the relevant activities of the function view as the information view for information management and the control of the material and information flow.

The Resource view represents the resources, i.e. human roles and technical agents as well as technological components of the enterprise as they are used in the course of enterprise operations. Resource models structure the assignment of resources to activities according to resource capabilities and need for that capability, e.g. for asset management.

The Organization view represents the responsibilities and authorities for all entities identified in the other views (processes, information, and resource) and provides the structure of the enterprise organization by establishing the identified organizational units into larger units such as departments, divisions, sections, etc.

The entity Model Content viewpoint model views of FIRO form the primary source of content from which other views of interest draw most of their content, i.e. the content of a different view most often results from a projection through the FIRO content filtering for specifics associated with the enterprise aspect that different view captures. As needed, other primary source modelling views may be defined and supported by the engineering tools, e.g. ecological, economic, and risk.

The entity Model Content view in particular has wide applicability because many currently used modelling languages generate models that fit in this category.

B.3.1.5.3.3 Entity purpose viewpoint

The following two aspect-oriented views allow representation of the model contents according to the purpose of the part of the enterprise entity that is being modelled.

- The Customer Service and Product view represents the model contents relevant to the enterprise's operation and to the results of that operation. This view represents the mission fulfilment of that enterprise.
- The Management and Control view represents the model contents relevant to management and control functions necessary to control mission fulfilment, i.e. that part of the enterprise that produces products or delivers services for the customer.

These aspect views delineate the scope of the enterprise, ensuring that the scope should extend to both the mission fulfilment part and the management part of the enterprise. An enterprise engineering methodology may propose preparation of separate models or descriptions for these two views.

B.3.1.5.3.4 Entity implementation viewpoint

Two different views, based upon the division between human and automated tasks, represent the means for implementation of the enterprise.

- The Human Activities view represents all information related to the tasks human roles are to accomplish. The view distinguishes between the tasks that can be done by humans, the extent of humanizability, and those that will be done by humans, the extent of automation, i.e. the limit of useful automation.
- The Automated Activities view represents all the tasks machines are to perform. This view includes information related to two distinct aspects: those tasks carried out by mission support technology and those tasks carried out by management and control technology, i.e. technology tasks. The implementation view distinguishes between the tasks which can be done by machines, the extent of automatability, and those which will be done by machines, the extent of automation, the limit of useful human endeavour.

NOTE Rapid innovation of technology across domains makes this viewpoint the primary source of information regarding trade-off opportunities involving labour and capital utilization (see [B.3.2.2](#)).

B.3.1.5.3.5 Entity physical manifestation viewpoint

Two different views allow representation of the physical manifestation of the enterprise.

- The Software View represents all information resources capable of controlling the execution of the operational tasks in the enterprise. These resources include any computer programme, either stored in a computer or in any other control device enabling the execution of an operational task or a set of instructions for human roles with defined skills such that the instructions for the humans enable performance of a task that they otherwise would not be able to carry out. Software can specify controllable states, e.g. a configuration description of manufacturing hardware such that the hardware in that configuration can perform a task provided that configuration is maintained for the duration of that task.
- The Hardware View represents all physical resources that have the capability to perform some sets of tasks in the enterprise. These resources include a computer system with given performance characteristics, an employee with given skills, or a machinery with given functionality.

B.3.1.5.3.6 GERA with aspect views

Figure B.10 shows an overlay of the different viewpoints and associated views identified above. Even though they share content, the aspect-oriented viewpoints are in general independent of each other, but certain combinations can be useful to represent specific aspects of the enterprise at specific life cycle phases. The availability of any view depends upon its implementation in the supporting engineering tool.

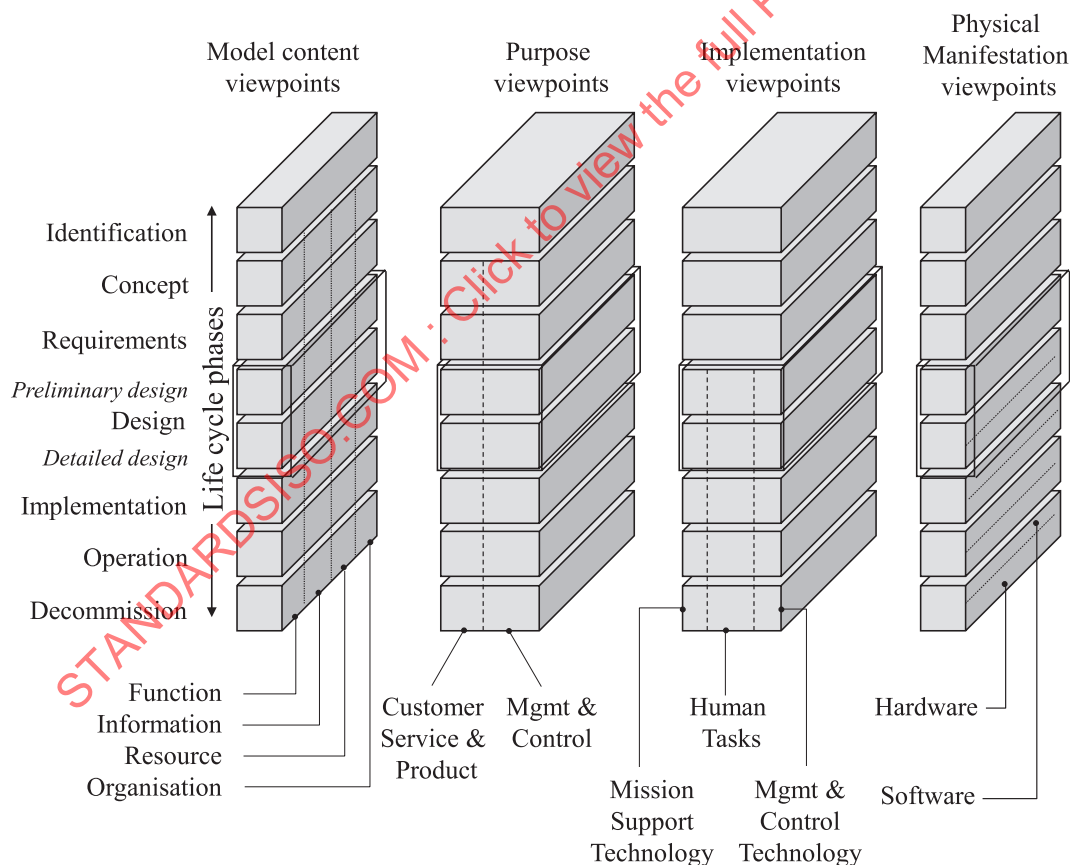


Figure B.10 — GERA Modelling Framework with Modelling Viewpoints and associated Views

B.3.2 Specification of EAMs

B.3.2.1 Methodologies for enterprise transformation

Enterprise architecture and integration methodologies describe the processes of enterprise transformation, such as the engineering of enterprise integration. The GERA life cycle concept identifies the scope for enterprise engineering methodologies.

A fundamental methodological aspect of Enterprise Integration/Enterprise Architecture (EI/EA) is that in fact EI/EA should not have a separate methodology in the sense of following a process that is completely separate from existing management and engineering processes. Rather, to ensure that there are no gaps or missing decisions, the EI/EA methodology needs to combine and harmonize existing:

- management processes and associated methods in strategic and tactical enterprise management, portfolio, programme and project management, as well as business analysis and alignment;
- associated engineering processes and methods.

For an organization to adopt an EI/EA methodology is equivalent to a systematic process improvement of its management and engineering practices, which improvement is a kind of transformation or enterprise integration endeavour, thus calling for a “bootstrapping” methodology to effectuate.

NOTE Creating a mature enterprise architecting practice requires the building up of the organizational hosts of governance, management and support functions in addition to the technical (architecting and evaluation) functions — and probably adopting standard process models for carrying out these functions.

By providing methods of progression for every kind of life cycle activity, an enterprise engineering methodology helps the user in the process of the enterprise engineering of integration projects whether in the overall integration of a new or revitalized enterprise or in management of on-going change. The upper two sets of these life cycle activities, identification and concept, are partly management and partly engineering analysis and description or modelling tasks. The requirements and design life cycle activity sets mostly involve engineering tasks throughout the process, including the production of enterprise models and designs.

Enterprise engineering methodologies describe the process of enterprise integration and guide the user in the engineering tasks of enterprise modelling. Different methodologies can exist that will cover different aspects of the enterprise change processes. These can be complete integration processes, or incremental changes as experienced in a continuous improvement process.

The enterprise integration process itself is usually directed to a repetitive service, or manufacturing enterprise or a project enterprise. The methodology may be specifically oriented to the type of enterprise or entity under consideration.

Enterprise integration may itself be carried out as a specific project or programme. But the integration task may begin at any of the enterprise's life cycle phases, not necessarily with identification or concept phase.

Therefore, in an enterprise engineering methodology, the processes relating to the different tasks of enterprise engineering should be defined independent of each other in order to allow for their combination in the context of the specific engineering task.

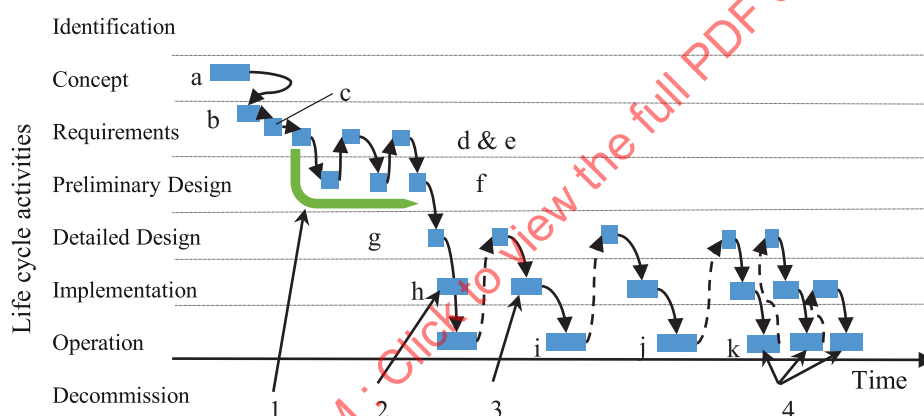
EXAMPLE An engineering project for a new production facility does not necessarily need to begin with the identification and concept definition phases of the plant because it is possible that the customer, who is commissioning the designing and building of the plant, has carried out these activities already. In this situation, the engineering project enterprise should only specify the requirements and carry out the design/detailed design, and implementation or building of the plant. Such an engineering project utilizes only the requirements, design, and implementation portions of a complete enterprise engineering methodology.

A large number of methodologies exist for enterprise transformation and engineering, including those popularized in the software and systems literature. While all these methodologies are based upon valid

ideas, none can be said to be the best, because the actual set of activity instances and their sequence will depend on the following:

- the type of entity that is being developed or transformed;
- the knowledge, skills, expertise and experience of the managers, analysts, architects and developers involved;
- the knowledge, skills, expertise and experience of the customers, both in technical and management terms, including clients and end users;
- the availability of good reference models;
- the availability of reusable components or products (modules in GERAM terminology).

An enterprise engineering methodology often has a distinctive pattern of life cycle activities evident in the associated life history. A simple waterfall methodology has a linear appearance of life cycle activities in the life history while an agile methodology has a repeating series of full waterfall like segments in the life history of activities as each new sprint occurs. Most methodologies have overlapping segments of activities occurring as iterations of design, implementation, and testing operation. [Figure B.11](#) depicts the life history pattern for an idealized application of the spiral development methodology in the context of a systems engineering V-model approach^[63].



Key

a	concept	g	detailed design
b	requirements	h	coding
c	analysis	i	unit test
d	requirements specification	j	system test
e	analysis	k	acceptance test
f	software design		

NOTE 1 The iteration occurring between Requirements and Preliminary Design and between Detailed Design and Implementation increase the extent of detail and certainty with corresponding risk reduction.

NOTE 2 Build is omitted, both for unit and integration testing and acceptance testing.

NOTE 3 Test-bed is needed to test both the unit and the system in operation.

NOTE 4 Depiction is omitted of the various alpha, beta and version releases.

Figure B.11 — Life history pattern of spiral development using system engineering V-model

Process models or descriptions with detailed instructions for each kind of activity of the integration process may describe enterprise engineering methodologies. This allows not only a better understanding of the methodology, but provides for identification of information to be used and produced, resources needed and relevant responsibilities to be assigned for the enterprise engineering process in the course

of project management of integration projects. A process representation of a methodology can employ the relevant enterprise modelling languages. Enterprise engineering methodologies may also use modelling methodologies as components. A modelling methodology is a methodology with the aim of giving help to model developers who use a modelling language or set of languages. It describes how a model can be developed and validated, either starting from scratch or using pre-defined partial models.

Typical contemporary enterprise integration/enterprise architecting methodologies, include: TOGAF Architecture Development Method (ADM)^[66], Unified Modelling Language (UML) 2.5 with scope of IT architecture development (see ISO/IEC 19505), Purdue Guide for Master Planning with scope of factory automation^[71], NATO Architecture Framework (NAF) 4.0^[51], Control Objectives for Information and Related Technologies version 5 (COBIT5) with scope of aligning enterprise strategy and IT architecture development^[41], Networked Enterprise^{[50][29]}, etc. In the future, such methodologies are likely to take into account or rely upon the standards for system engineering life cycle processes (ISO/IEC/IEEE 15288), architecture development processes (ISO/IEC/IEEE 42020) and architecture evaluation (ISO/IEC/IEEE 42030).

An EAM may also be considered to provide partial process models for relevant management and engineering functions of enterprise entities involved in enterprise transformation. From a deployment aspect, the processes, activities and tasks that implement these functions can be distributed in various ways among roles in a particular enterprise.

B.3.2.2 Human factors

The major part of a methodology is a structured approach that defines not only all the steps/phases to follow in an engineering and/or integration project, but also the way of involving as much as possible the people working in the company, the eventual users, in the analysis and design of the manufacturing and service system.

The involvement of internal stakeholders is an important success factor for an integration project. Techniques used to build new manufacturing and service systems are still difficult to understand for business users of the future system, particularly in the domain of the information technology. Through continuing involvement prior to development and implementation, these stakeholders can verify that the design solution of the new or improved system will meet their needs and expectations, including investment constraints, and will result in successful validation of the system in operation.

The involvement of people of the company facilitates final acceptance of the designed system and thus shortens the transition time between the old and new systems. The methodology should make clear distinction between the two major design efforts: user-oriented design and technology-oriented design. Care is necessary when engaging people to assure suitable competency and skill in the various design efforts.

B.3.2.3 Assigning tasks to human roles and machines

Another critical aspect of human involvement in the enterprise is the role of humans in the enterprise, i.e. which tasks do they perform and how do they interact with automation. For most enterprises, human involvement continues its accelerating rate of change and understanding the task relationships between human roles and machines is essential.

To understand the role of humans in the implementation of enterprise functions, one approach is to assign the appropriate tasks and functions developed in the Requirements modelling life cycle phase to the human role element of the system by first considering the functional tasks grouped into three categories at the Preliminary Design modelling life cycle phase (see [Figure B.12](#)).

This assignment separates the tasks of enterprise Mission Fulfilment and Enterprise Management and Control, as defined during requirements analysis, into

- the automated information tasks that become the Management and Control Information Systems Architecture functions,

- the automated manufacturing and service tasks that become the Mission Support Equipment Architecture functions, and
- the remainder of non-automated tasks becoming the functions carried out by humans as the Human and Organizational Architecture.

The split of functions for implementation between humans and machines forms the first definition of the implementation of the resulting system. Three separate elements in the implementation scheme are necessary because while humans participate in all three elements, automation participates in only two of these elements. This distinction is illustrated in [Figure B.12](#) where dash lines represent the limits of automation and the limits of human involvement where

- the Extent of Automatability boundary is the absolute extent of technologies in their capability to currently automate the tasks and functions because many tasks and functions require human innovation, human accountability, human ability to handle unexpected situations, etc., and cannot be automated with then available technology, and
- the Extent of Humanizability boundary is the maximum extent to which humans can be used to implement the tasks and functions because of human abilities related to speed of response, breadth of comprehension, range of vision, physical strength, etc.

Another kind of boundary, the Extent-of-Automation, is the extent of automation carried out or planned in the enterprise. The Extent-of-Automation, indicated by dotted lines in [Figure B.12](#) defines

- the boundary between the Human and Organizational Architecture and the Management and Control Information System Architecture, and
- the boundary between the Human and Organization Architecture and the Mission Support Equipment Architecture.

Each of these four boundaries can fluctuate frequently as automation capabilities advance and the economics of human labour resources change.

As long as requirements are fulfilled, e.g. timing, co-ordination, cost, risk, it makes little difference whether humans or machines carry out a specific function or task, or which utilization of organizational structure or human-relations requirements occurs. Therefore, political and human relations-based considerations as well as technical constraints determine the actual extent of automation. Economic, political, social, i.e. customs, laws and directives, union agreements, etc., as well as currently available technology all influence the relative position of these Extent of Automation boundaries.

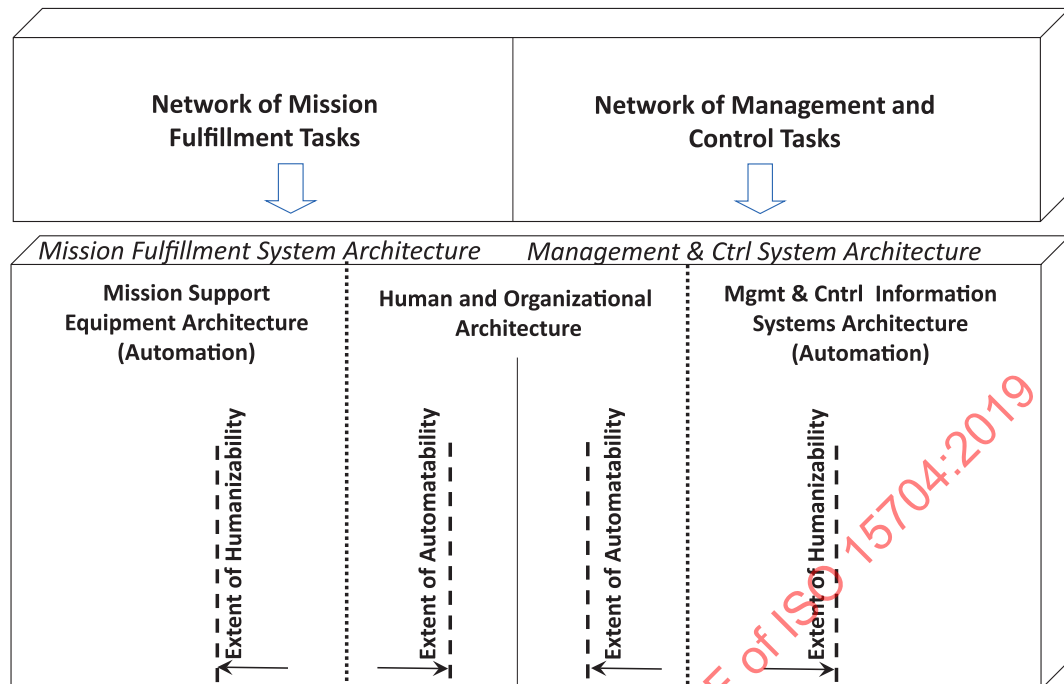


Figure B.12 — Boundaries of Automatability, Humanizability and Extent of Automation for defining three Implementation Architectures

B.3.2.4 Project, programme and portfolio management

In order to perform the analysis, the design and the implementation within an engineering and/or integration project efficiently, the methodology needs to align with the available project and programme management techniques in terms of planning, budgeting and control, follow-up, etc.

A logical separation can be made between a Project life cycle and Enterprise System life cycle (see [B.3.1.3.4](#)). Within the project or programme life cycle, and guided by the various phases defined in the life cycle of the system that is designed/built by the project,

- the “management and control” part of the project life cycle covers project management, and
- the “service to the customer” part covers the execution (operation) of the project.

In this sense one of the main activities within the project or programme management's operation is the planning of time and resources and the control of the steps to be executed and defined in the system life cycle. In the case of a programme, or of a very large project that has many sub-projects, the main effort is spent on designing, establishing and coordinating these sub-projects, and establishing a close relationship between strategic goals of the enterprise and programme and project mandates.

The life history of a project or programme contains at least these three stages in time:

- a) start-up, aimed at defining the organization (various teams and managers), preparation (definition of the what, who, when and how), planning and the structure of the project or programme;
- b) mission fulfilment, where project operations fulfil the project's mandate, under the guidance of project management and control, aimed at acceptance of deliverables (hard and/or software, machines, various installations...), monitoring of progress and continuous planning, managing problems and change, contracting suppliers, executing reviews, auditing, etc.;
- c) wind-down and dissolution, aimed at the general acceptance and the final evaluation of the project.

Widely used contemporary management methodologies for project management include among others: for project management, the Project Management Body Of Knowledge (PMBOK)^[58] and Projects In

Controlled Environments version 2 (PRINCE 2)^[57]; for programme management the Programme Management^{[56][59]}; and for portfolio management^[42].

B.3.2.5 Economic and performance aspects

A methodology needs to take the economic aspect into consideration. In fact, the choice of various investments depends upon objectives that are often contradictory. To help designers choose the best solution, an examination of both technical and economic views at the different steps of an integration project is necessary.

The methodology should allow the decomposition of the strategic goals of the company into objectives related to each function; and a technical-economic evaluation needs to follow the specification of the technical solution. The economic evaluation can be partitioned into these three steps:

- a) calculate the cost of the solution;
- b) measure performance of the solution;
- c) compare the solution costs with the budget.

The aim of this approach is on the one hand, to compare the project cost against the investment budget, and on the other hand, to compare the solution performances against the technical objectives derived from the company strategy. This comparison informs a determination of the economic viability of the proposed solution.

Technical-economic evaluation approaches can be found in ECOGRAI^[35], GEM (GRAI Evolution Methodology) and Activity Based Costing^[32] among others.

B.3.2.6 Other aspects

If intended for applicability of the methodology, an enterprise engineering and architecting methodology may specify a number of other aspects deemed necessary to design for and evaluate depending on the domain, e.g. risk elements of security, and safety.

B.3.3 Specification of EMLs

The engineering of an enterprise is a highly sophisticated, multidisciplinary management, design and implementation effort during which various forms of descriptions and models of the target enterprise are created.

Enterprise models need to represent the enterprise operations from various modelling viewpoints, which may specify different modelling approaches to satisfy the collection of stakeholder concerns. The set of languages utilized needs to be capable of expressing the models of all areas defined in the modelling framework of the GERA. For each area of the GERA modelling framework, the enterprise engineering methodology may select a different modelling language that is suitable for the expression of the respective models. In practice, the set of languages will be smaller than the set of areas to be modelled, with one language suitable for more than one area.

A complete set of enterprise modelling languages needs to satisfy the following:

- every area represented in the modelling framework (see [Figure B.8](#) and [Figure B.10](#)) needs to be covered for every enterprise entity type;
- a model developed in one subject area needs to be able to be integrated with models of other subject areas, if the information content of the model so requires.

More than one modelling language may cover any subject area of modelling because the languages have different expressive power, e.g. it is possible that a language is useful only for the description of the subject area but not suitable for certain analysis tasks, or the languages that belong to the function view can differ in their capability of expressing certain characteristics of functions: dynamics of the function, behaviour of the function or the subdivision of the function into function types such as product