INTERNATIONAL STANDARD

ISO 15704

First edition
2000-06-01

AMENDMENT 1
2005-08-15

Industrial automation systems — Requirements for enterprise-reference architectures and methodologies —

AMENDMENT 1: Additional views for user concerns

Systèmes d'automatisation industrielle — Prescriptions pour architectures de référence entreprise et méthodologies —

AMENDEMENT 1: Vues additionnelles pour les intérêts de l'usager

Reference number
ISO 15704:2000/Amd.1:2005(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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.

Amendment 1 to International Standard ISO 15704:2000 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 5, *Architecture, communications and integration frameworks*. In preparing this amendment, substantive contributions were received from groups involved with enterprise-reference architectures such as the Purdue Enterprise-Reference Architecture (PERA), the Graphes et Résultats et Activités Interreliés GRAI Integrated Methodology (GRAI GIM), the Computer Integrated Manufacturing Open System Architecture (CIMOSA), and the Generalised Enterprise-Reference Architecture and Methodology (GERAM).

Industrial automation systems — Requirements for enterprise-reference architectures and methodologies —

AMENDMENT 1: Additional views for user concerns

Page vi, Foreword

Replace the last paragraph with the following:

"Annexes A, B, C, and D are informative. Annex A is based on version 1.6.2 of GERAM developed by the IFIP/IFAC Task Force on Architectures for Enterprise Integration who granted permission for its inclusion in ISO 15704. Annex B is based on the economic view found in A Stair-Like CIM System Architecture of Chen and Tseng. Annex C is based upon the decisional view found in CEN/TS14818 Technical Specification – Enterprise Integration – Decisional Reference Model."

Page 1, subclause 3.2

Replace (a) in the note with the following:

"a) system architectures (sometimes referred to as "type 1" architectures) that deal with the design of a system, e.g. the computer control system part of an overall enterprise integration system;"

Page 5, subclause 4.2.6

Replace with the following:

"Enterprise-reference architectures and methodologies shall exhibit the capability to represent any process and its constituent activities for the accomplishment of the management and control in support of the established mission of the enterprise according to the criteria established by enterprise management."

Page 6, subclause 4.2.10

Add the following paragraph after the last paragraph:

"Model developers may generate additional views for particular user concerns, and these can then be used by any concerned stakeholder. Examples of additional views are found in annexes B and C."

Page 41, annex B

Add the following two annexes before the existing Annex B and renumber the existing Annex B and its subclauses accordingly.

Annex B
(informative)

Economic View in CIM system architecture

B.1 General

B.1.1 Introduction

For entrepreneurs and business managers, confidence in advanced CIM technology depends upon the realization of a return on investment projected from design phase activities of both new system implementations and system up-grades and re-organizations/integrations. Since both tangible and intangible benefits must be considered, evaluating the return is a difficult problem. An essential aspect of any mechanism to resolve the problem is the ability to evaluate different alternatives using models of existing and proposed system architectures in a manner that connects functionality with economic consequence so that design trade-off decisions are possible. In particular, the evaluation of intangible benefits is often a barrier to Computer Integrated Manufacturing investments.

An Economic View presents model content relative to economic decisions. It draws upon existing model content and established analytical methods to inform decision makers. The view is most critical early in the life cycle when the majority of economic commitments are encountered and late in the life cycle when economic performance is measured.

B.1.2 Support for enterprise managers

As guidance for enterprise managers, the Economic View can help them to:

a) predict the influences of system integration on the enterprise,

b) evaluate necessary investment and possible benefits,

c) make decisions and improve their correctness, and

d) monitor the implementation process and application of the integrated system.

B.1.3 Support for enterprise model developers and analyzers

As guidance for model developers and analyzers, the Economic View helps them to:

a) describe the economic elements,

b) understand relationships between these elements and other components in an integrated system,

c) describe economic relationships among enterprise strategic targets, the framework of the integrated system and its components, and

d) identify economic benefits of enterprise re-organization.

B.1.4 Support for system developers

As guidance for the system developers, the Economic View provides:

a) methods to evaluate economic consequences of system function modifications during the system development, and

b) scoping of software tool use for economic modeling and analysis.

B.2 Framework for Economic View

In system implementation/integration projects, the goals and corresponding demands of the project target are reflected in the demands of the economic characteristics. Their economic implications/influences on the system are realized through the integration strategy and the technology project. The Economic View establishes the relations between the economic target and the engineering project. It describes economic elements, influence factors and scalar indices manifested in the integrated system and their relationships that allow the determination of their impacts on the economic targets in the system integration project. These indices, factors, and elements are constructs and their properties taken from or derived from the four mandatory model-content views (4.2.10)

In an integrated system, the Economic View consists of a grouping of models, which is used to describe economic components and their relationships. Many methods, e.g., graphical, mathematical, and even descriptive ones, may describe economic components. In order to improve the compatibility and assure the successful operation of an enterprise, a three-layer framework is constructed, expressed in graphic form, based on enterprise modeling methods and reference models in the general enterprise reference architecture, as shown in Figure B1.

The three layers in the Framework for Economic View (Indices, Factors, Elements) possess different economic attributes and the relationships among layers have different attributes as well. The framework establishes the relationships between layers of detail from the top level strategic targets of an enterprise to the bottom basic economic elements with intervening indices and factors. To correctly establish the relationships among different layers, both clustering and classification methods should be used to gather information from the generic and partial model pool for the applicable life cycle phases and then classify the information to establish the particular trees and relationships.

Early in the life-cycle, economic targets (ET) and constraints are established, e.g., return on investment, and pricing levels. Relative to this domain identification and concept definition, sets of economic indices (I_J) bearing on the targets and constraints are arranged and analytic methods are chosen with increasing levels of detail exposed as the life cycle progresses. At the factors layer, process related cost factors are derived from the decomposition of process models into activities (f_P). At this layer other economic factors result from the analytical breakdown of expected value that can be both tangible and intangible (f_A). All of the indices have both tangible and intangible factors. Even the most tangible indices, cost (I_C) and time (I_T) may have intangible factor influences that need to be taken into consideration. The explicit intangible factors, service (I_S) and environment (I_E) may have tangible factors as well, e.g., response time, pollution rate, etc. Tangible factors have diverse forms and representation. They can be expressed in mathematical equations (f_E), matrices, tables (f_T), boxes in graphical models, etc. In Figure B.1, the design phase is shown with greater elaboration using a tree of decomposed indices, process factors (f_P) depicted as a process model fragment, analytical factors (f_A) depicted as hierarchy models, equation factors (f_E) depicted as a formula, and table factors (f_T) depicted as a data table.

For factors, the element layer identifies the basic economic elements that comprise the variables in the mathematical equations (ϵ_E), the entries in the matrices and tables (ϵ_T), the activities (such as an activity box, e.g., in the lowest level IDEF3 model, (ϵ_A)), etc., from which the factor cost or value are derived. These elements are usually simple attribute values characterized as indivisible, and can be used to measure, monitor, or control the related factors. In general the elements are properties of resources used to value and cost an activity.

Economic indices, factors, and elements can be of generic types collected as a pool of constructs for use at the various layers. These generic types can be formed into partial models of indices and factors to be used as an aide for populating a particular economic view through specialization.

Analysis methods vary by layer with, for example, tree hierarchy analysis techniques appropriate at the Indices layer, and process structure model simulation, hierarchy analysis, physics formulas, fit and interpolation methods at the Factor layer. These analysis methods collect data and support the decision optimization of the enterprise. Optimization results can be imposed on attributes to realize the enterprise strategy and improve its competitive ability. Two iterations of optimization and control exist - the target decomposition from the top down at Requirements, followed by system analysis from the bottom up at Design occurs early in the life cycle and then the system implementation from the top down at Implementation and the system monitor and control from the bottom up at Operation occurs later in the life cycle. The first iteration results in the roll-up of economic valuations for comparison against the targets and constraints. The second iteration provides measures of economic performance.

Such methods can assure the realization of the enterprise target, the fundamental information collection and analysis, the rationalized target fulfillment and the system monitoring. Implementation of the framework should be supported by correct methodology, rich engineering practices and advanced theories and methods of system integration. Initiatives in concurrent engineering, cell technologies, and total quality management may be coupled with capital and labor investment for economic benefit.

The analysis and evaluation of different implementation alternatives of CIMS can be performed using the Economic View. The selection of the best alternative from many opportunities to implement system integration and the improvement of the enterprise competency is achieved as a result of specific modeling methods.

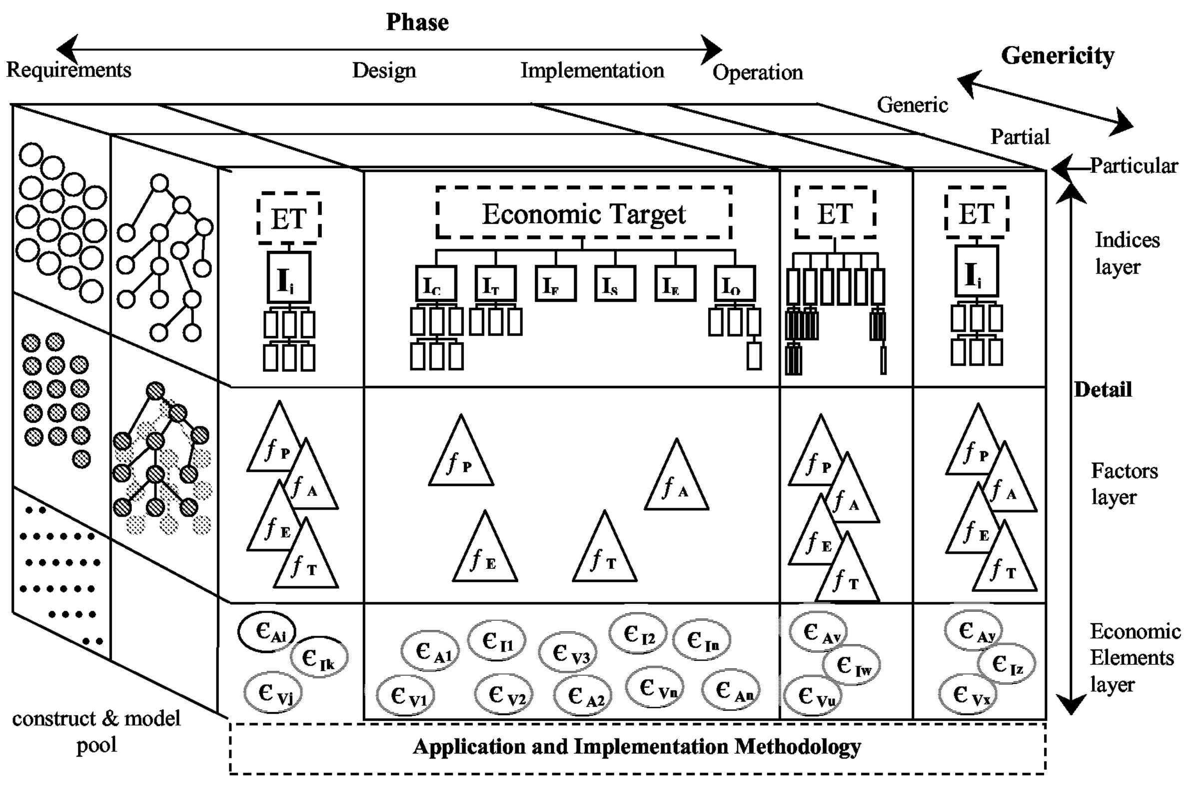
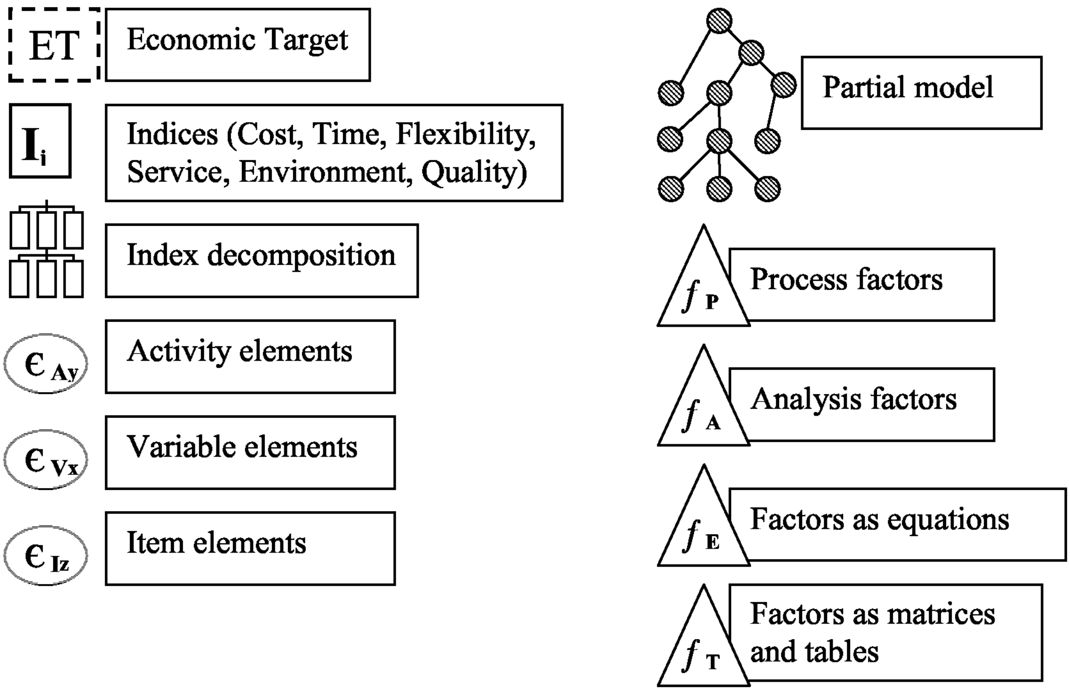


Figure B.1 — Framework for Economic View

Table B.1 — Icons for Figure B.1



B.3 Candidate modelling methods

B.3.1 Introduction

Two methods used at the Factor layer, depicted in Figure B.1 as f_A and as hierarchy models, are presented below and followed by illustrative examples.

B.3.2 Activity Based Costing

Activity Based Costing (ABC) is a method to measure the cost and performance of an organization based on the activities, which the organization uses in producing its output. ABC differs from traditional cost accounting techniques in that it accounts for all "fixed" and indirect costs as variables, without allocating costs based upon a customer's unit volume, total days in production or percentage of indirect costs. Information gathered through ABC should provide a cross-functional, integrated view of your organization, including its activities and its business processes. [1]

B.3.3 Analytic Hierarchy Process/Analytic Network Process

The Analytic Hierarchy Process (AHP) is a decision making process to help set priorities and make decisions when both qualitative and quantitative aspects of a decision need to be considered. By reducing complex decisions to a series of one-to-one comparisons, then synthesizing the results, AHP helps decision makers arrive at optimal decisions and provides a clear rationale for those decisions. The AHP engages decision makers in breaking down a decision making procedure into smaller parts, proceeding from the goal to criteria and sub-criteria from the Indices layer, down to the alternative courses of action. Decision makers then make simple pair wise comparison judgments throughout the hierarchy to arrive at overall priorities for the alternatives. The decision problem may involve social, political, technical, and economic factors. The AHP method helps people cope with the intuitive, the rational and the irrational factors, and with risk and uncertainty in complex settings. It can be used to: predict likely outcomes, plan projected and desired future, facilitate group decision making, exercise control over changes in the decision making system, allocate resources, select alternatives, do cost/benefit comparisons, evaluate employees and allocate wage increases. [2]

The Analytic Network Process (ANP) is a general theory of relative measurement for deriving composite priority ratio scales from individual ratio scales that represent relative measurements of the influence of attributes that interact with respect to control criteria. Through its super matrix, whose attributes are themselves matrices of column priorities, the ANP captures the outcome of dependence and feedback within and between clusters of attributes. The Analytic Hierarchy Process (AHP), with its dependence assumptions on clusters and attributes, is a special case of the ANP. ANP augments the linear structures used in traditional approaches and their inability to deal with feedback in order to choose alternatives. ANP offers decision making according to attributes and criteria as well as according to both positive and negative consequences.[3]

B.4 Applying Economic View in model development

B.4.1 Introduction

Using the candidate methods of B.3, a subset of an Economic View as an example is presented below. The models chosen help decision makers align costs and value with targets and constraints.

B.4.2 ABC Method example

In order to accurately assess CIM technology benefits to enterprises, a costing technique that considers not only production but also other processes is required. For this example the modeling formalism is based on the IDEF0 method.[4] Since both ABC and IDEF0 focus on functional activities, the IDEF0 model is extended to include activity based costing data. In this way we assure that no activity cost assignment will be missed during the integration with an IDEF0 model. Here, a separate economic model that corresponds to the IDEF0 model of function view is constructed. There are four attributes in each model block: 1) node number, 2) activity name, 3) cost driver and 4) cost value. The first two attributes are taken directly from an IDEF0 model, whereas the latter two are to be defined by designers. As shown in Figure B2., the cost model forms a hierarchy exactly like the IDEF0 model. Sub-processes are defined down to Element layer activities that are the most basic.

Guidelines for constructing an ABC economic model include:

a) No attribute can be left empty;

b) Cost value of a parent process must be the sum of the cost values of all its lower-level sub-processes or activities;

c) If there is a cost for coordinating activities of the same level, coordination should be modeled as an activity of that level;

d) The model can be decomposed as a hierarchy equivalent to the IDEF0 hierarchy;

e) Assignment of cost values should be done in a bottom up manner, so that higher-level activity cost values can be consolidated and assigned accordingly.

For example, as shown in Figure B.2, the cost drivers of the process ‘Delivery of Part As’, ‘Preparation of raw material’, ‘Production of Part As’, ‘Purchasing material’, ‘Work order control for part delivery’, ‘Preparation of NC program’, ‘Machine set-up’, and ‘Machining’, are defined. Then we assign cost values for ‘Preparation of NC program’, ‘Machine set-up’ and ‘Machining’ (basic economic Elements). Hence, the cost value for ‘Production of Part As’ is calculated by summing the A2 cost values (A21 + A22 + A23). Similarly, the cost values for ‘Preparation of raw material’, ‘Purchasing material’, and ‘Work order control for part delivery’, are assigned. Finally the cost for ‘Delivery of Part As’ is determined. In order to deliver a product, processes like production planning and shipping are necessary and thus the costs for these processes are added to determine the total cost of a product. Note that the ABC modeling method can be applied to the existing processes as well as estimating costs for new systems. The objective is to accurately capture or estimate the project costs.

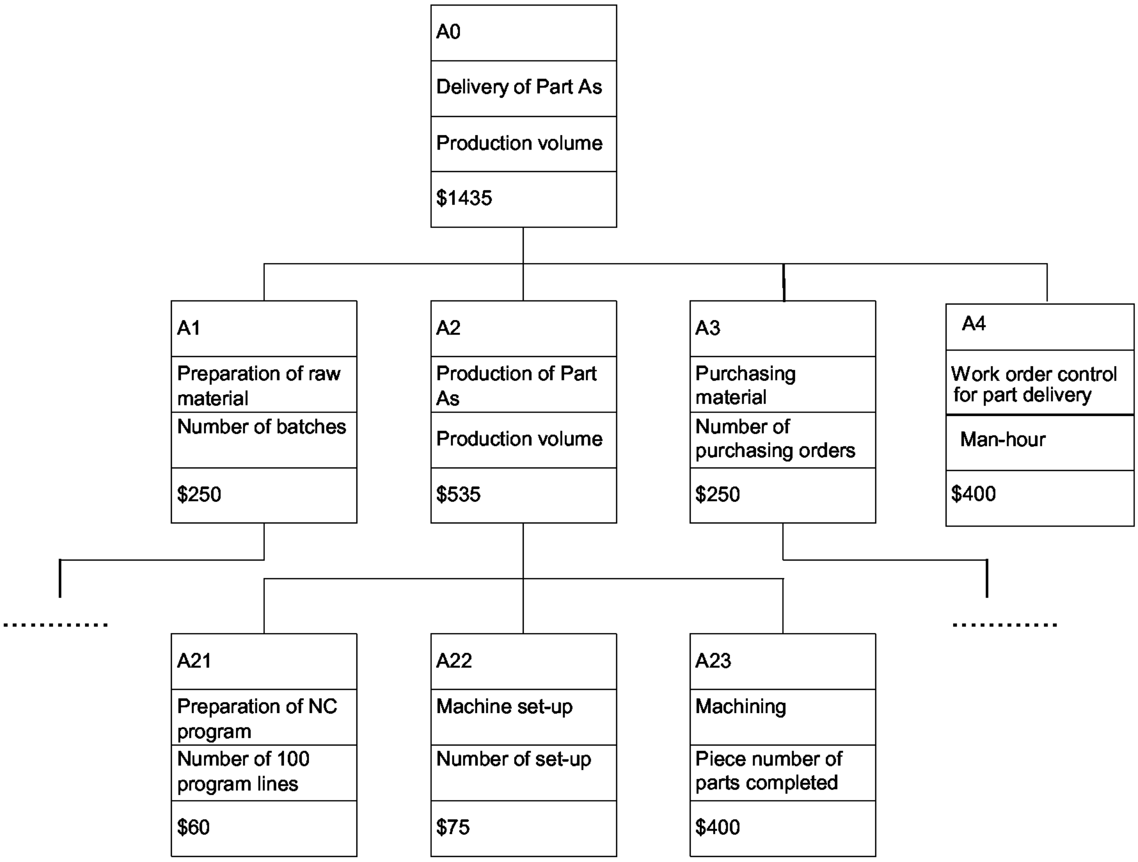


Figure B.2 — Example of a cost hierarchy

B.4.3 AHP Method example

Since investing in CIM often is not for the sake of the technology itself, it is especially important that the resulting business and manufacturing processes meet the target performance. Operational measures of performance should be derived from company goals that align with corporate strategies at Indices layer. The questions to resolve are: 1) whether the technology investment can effectively bring the business to the target, and 2) is the investment economically sound. The Activity Based Costing technique discussed in the above section (B.3.3) addresses the tangible aspect and deals with the second question. The first question is addressed using the Analytic Hierarchy Process (AHP) method at the Factors layer.

For example, a manufacturing company is launching a technology advancement project in order to keep company growth on target. Funds are reserved for the first stage of project effort. Due to a budget limit for the first phase, a team of managers, analysts and engineers are asked to make an investment proposal. The AHP method is employed by the team to decide which area of the project will receive initial funds allocation. A hierarchy of the advancement investment problem is constructed as in Figure B.3.

During the analysis, it is observed that product cost, production lead time, product quality and customer service contribute differently to market share and profitability. Similarly, increasing market share and enhancing profitability are contributing differently to the goal of company growth. The Analytical Hierarchy Process method weights the contributions of alternatives to the goal.

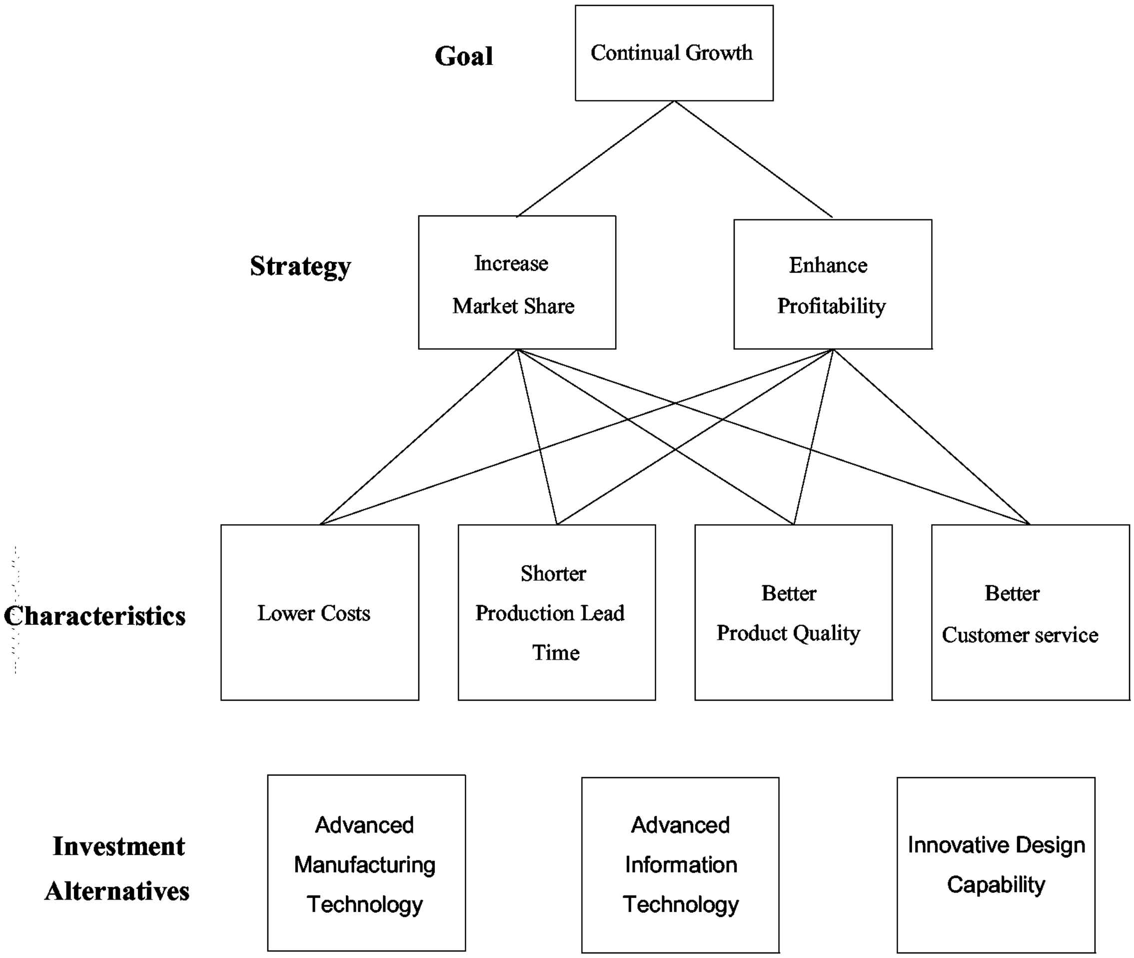


Figure B.3 — The hierarchy of advanced investment

B.4.4 Using example method results

In terms of cost and benefit analysis, benefit indices are defined based upon AHP priorities. The cost indices are defined using the ABC method. First, cost components of investment in manufacturing technology, information technology and design technology are determined. These cost components should include the process costs after the particular technology is invested as well. To reduce the possible bias caused by high capital costs, the capital cost may be left to the return on investment calculation. After the IDEF0 hierarchies and the cost hierarchies are built, the total cost is computed.

B.5 Glossary of references for Economic View

[1] Chen Yuliu, Tseng M M, Yien J. Economic view of CIM system architecture. Production Planning and Control, 1998, 9(3):241-249

[2] Saaty, T. L., Multicriteria Decision Making: The Analytical Hierarchy Processes, (McGraw-Hill), 1980.

[3] Saaty, R. W., Decision Making in Complex Environments, (Super Decisions),2003.

[4] National Institute for Standards and Technology, Standard for Functional Modeling – IDEF0, FIPS Publication 183, 1993.

Annex C
(informative)

Decision View of an enterprise model

C.1 Introduction

An enterprise is organised by functions and levels of responsibility. Decisions are made within multiple functions and multiple levels. The decision view is intended to support integration from a decision-making viewpoint. Decisions made within various functions shall be consistent in the sense that they shall contribute to achieving the global objectives of the enterprise. This also means that the time horizons in which various decisions are made are coordinated. In the domain of production management and control, to get the correct raw material/product at the correct time, on the correct machine and processed by the correct person implies that decisions are made in multiple time horizons.

The decision view described in this annex identifies production planning and control decisions and the relationships between them. These decisions are made using content from information and resource views under responsibility established in the organisation view.

The decision view is concerned with the description of an enterprise decision-making structure which provides for identification of decision topics, their categories, criteria and dependencies. This annex presents basic concepts relating to the decision view and focusing on the Production Management domain.

The decision view defines a generic integrated decision system structure in terms of a set of decision centres and decision links. It is a common structure for integrated decision-making in the domain of production planning and control. It serves as a basis to elaborate the decision model of a particular system.

The decision view is intended for those who are

a) decision-makers responsible of production management and control,

b) Involved in performing daily production planning and control activities,

c) Involved in designing production planning and control systems,

d) Involved in developing production planning and control software (i.e. MRPII, ERP, etc.), or

e) Involved in enterprise engineering and integration projects in general.

C.2 Decision View concepts

C.2.1 "Decision"

The term “decision” relates to “those activities or processes that are concerned with making choices”; the decision itself is “the result of choosing between different courses of action”. The activity of making a decision consists of choosing from amongst a set of known variables; the variable which best meets the objective, within the constraints.

C.2.2 Functional decision categories

Decision-making activities are classified into functional categories depending on the things they handle [Products (P), Resources (R) and Time (T)]. Combinations of these things lead to a categorization as follows (also see Figure C.1):

a) "manage products" (e.g. finished goods, sub-assemblies, parts and raw materials). These decisions are concerned with the management of products in time, ($P \cap T$). Major decisions of this category are concerned with what, when and in which quantity those products are to be procured and which levels of inventory are appropriate;

b) "manage resources" (e.g. information technology and manufacturing technology resources as well as humans). These decisions are concerned with the management of resources in time, ($R \cap T$). Major decisions of this category are concerned with the resource capacity/capability management;

c) "plan production" (e.g. master schedule, shop floor scheduling, etc.). These decisions are concerned with production planning that synchronizes the flow of products through resources in time, ($P \cap R \cap T$).

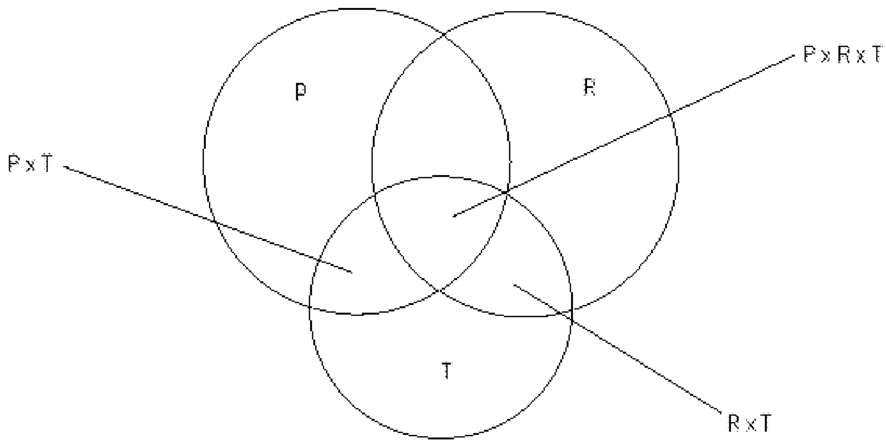


Figure C.1 — Three basic decision-making domains

C.2.3 Time decision categories

Decisions are classified into three general time categories:

a) Long-term and strategic in scope — these are long-term decisions that are concerned with the definition of objectives consistent with the global objectives of the enterprise;

b) Medium-term and tactical in scope —. these are medium-term decisions that deal with the implementation of means (both human and machine resources) to meet the strategic objectives;

c) Short-term and operational in scope —. these are short-term decisions associated with planning and execution of actions, using the means defined at the medium-term.

C.2.4 Specific time decision concepts

C.2.4.1 Horizon

A horizon is the part of the future taken into account by a decision, i.e. the horizon is six months when a decision is taken on a time interval of six months. The concept of horizon is closely related to the concept of planning. Thus the concept of horizon is also very close to the notion of time category (long-term, short-term, etc.) but is more precise. For example, in industrial production systems, a horizon gets quantified in relation to the customer order lead-time, the material requirements cycle times and the manufacturing cycle.

C.2.4.2 Period

The concept of period is closely related to the concept of control and adjustment. When a decision, based upon an objective, has been made to carry out some activity or activities during a subsequent horizon, the execution of these activities needs to be monitored. The intermediate results need to be measured with respect to the objective before this activity is completely finished and the horizon has terminated. If the measurements show that there is a deviation with respect to the reference objective, adjustments should be made. A period is the time that passes between a decision and when this decision shall be re-evaluated.

EXAMPLE A three-month plan may be re-evaluated and decided upon every two weeks, e.g. the horizon is three months and the period is two weeks.

The concept of period allows a manager to take system changes into account. These changes can come from the internal behaviour of the system (for example disturbances or machine breakdowns) and from outside (for example new customer orders arrive or problems arise related to providers).

C.2.5 Decision level

Decision level is an abstract concept that represents a decision-making hierarchy. It is defined by a pair of values indicating a horizon and period, (H,P). At a given decision level, all decisions made will have the same pair of values for horizon and period.

A particular decision level may be mapped to one of the three basic time categories (long-term, medium-term, and short-term). Each of the three basic levels may be decomposed on sub-levels. For example, in some companies, the long-term level may have two sub-levels with decisions dealing respectively with manufacturing strategies and long-term production planning. (see C.4.)

C.2.6 Decision centre

Decision centre is an abstract concept that represents the intersection of a decision level and a functional decision category of the domain. Decision centres are mapped onto an enterprise organization to identifying the people responsible for making various decisions. A decision centre is defined as the set of decisions made at one decision level and belonging to one functional category. Decision centres are the conceptual locations where decisions are made about the various objectives and goals that the system should reach and about the means available to operate in accordance with these objectives and goals. To manage a system, many decision centres operate concurrently, each with its own dynamics reflecting the various time-scales and dynamic requirements that management decisions need to address.

C.2.7 Decision frame

C.2.7.1 Decision frame content

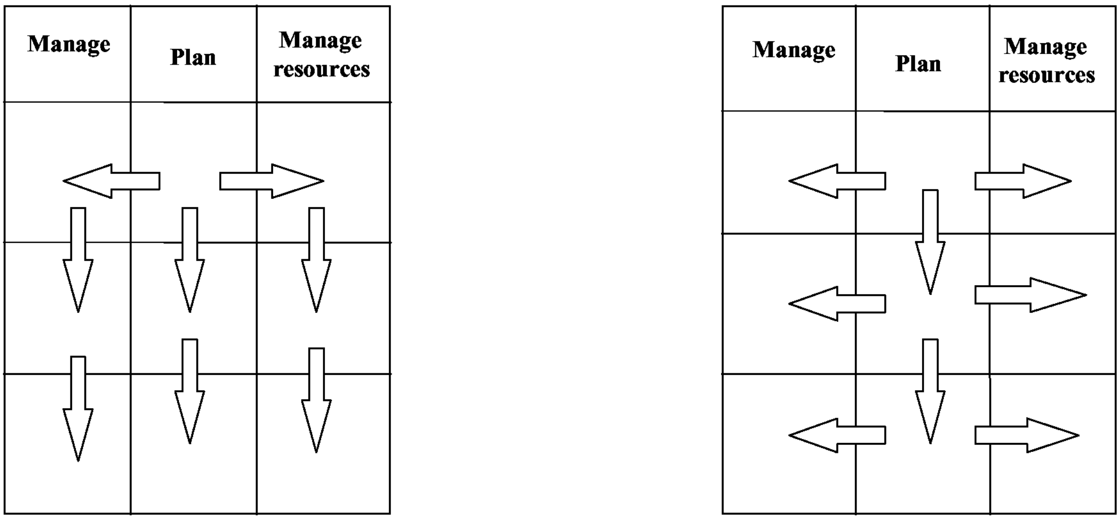
A decision frame is composed of the information content transmitted between decision centres that describes a set of items constraining the degrees of freedom for decision-making (see Figure C.2). This frame shall not be modified by a decision made in the decision centre receiving the frame. To avoid conflicts, a decision centre should be under the influence of only one decision frame.

The main items influencing decision-making are:

a) the decision objective or set of objectives that the decision has to meet;

b) the decision variables that enable the decision-maker to know the scope of available actions and their constraints;

c) the decision criteria that guide the choice in decision-making.



Coordinating structure **Synchronizing structure**

Figure C.2 — Styles of decision links

The items that make up the frames of decision-making are primarily determined by the hierarchy of the decision system. These items are decomposed in a consistent way by descending through the hierarchy of the system. Hence, through the decision frame, a decision centre transmits to another decision centre the objectives, the decision variables, the constraints and criteria that the latter should take into account when making decisions. This transmission is conducted as a dialogue referred to as a decision link.

There exist two basic frame structures defined by different styles of dialogue: coordinating and synchronizing as shown in Figure C.2. The coordinating structure emphasizes the coordination between various decision levels, while the synchronizing structure emphasizes the synchronization between various functional decision categories. The choice of structure depends on the management style of the enterprise and the situation for which the grid is being applied.

C.2.7.2 Decision objective

Objectives indicate performance targets. These performance targets can be the production costs, the delivery lead-time, or the level of quality, for example. Objectives are needed at every decision centre each time a decision is made. Global objectives refer to the entire production system and, according to the principle of coordination, are consistently detailed to give local objectives to all decision centres.

C.2.7.3 Decision variable

Decision variables are the items upon which a decision centre can make decisions that allow it to reach its objectives.

EXAMPLE For scheduling workers' working hours, a decision variable can be "the number of extra work hours", e.g. the decision frame of scheduling declares that scheduling decisions may decide upon the value of extra working hours in order to reach the objective of scheduling.

A decision centre may act upon one or more decision variables through determining their respective values. In other words, decisions are made in a decision space. The dimensionality of a decision space is determined by the number of decision variables.

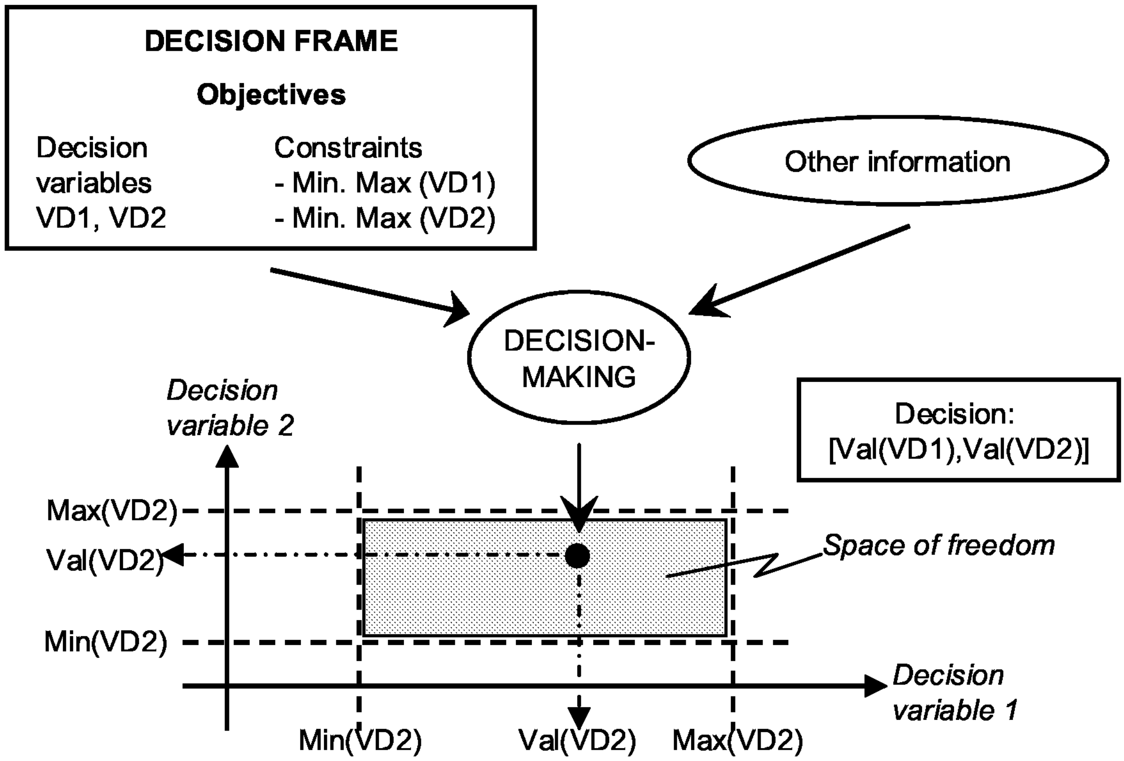


Figure C.3 — Decision space within a decision centre

C.2.7.4 Decision constraint

Constraints are the limitations on possible values of variables. Decision constraints limit the freedom of a decision centre to select any arbitrary value for its decision variables.

C.2.8 Performance indicator

A performance indicator is an aggregated piece of information that provides a measure allowing the comparison of the system's performance to the system's objectives. A performance indicator is defined by its name, a value domain and a procedure for determining its value.

Performance indicators should be consistent with objectives because it is necessary to compare performance targets (objectives) and performance attainment (indicators). Performance indicators should also be consistent with decision variables because those variables will have an effect on the performance monitored (controllability). The main issue is to ensure internal consistency within a decision centre in terms of the consistency triplet presented (see Figure C.4). This consistency is ensured if the performance indicators allow verification of the achievement of the objective and are influenced by actions on decision variables.

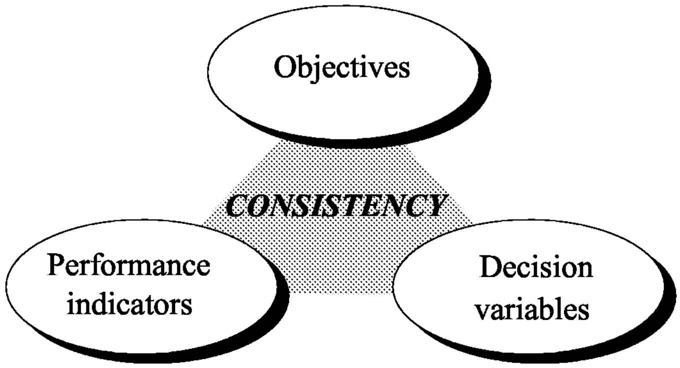


Figure C.4 — Consistency of the {Objectives, Variables, Performance Indicators} triplet

C.3 View formalism

The decision view is represented by a grid (see Figure C.5). Rows represent decision levels and columns signify decision functional categories. The intersection between a row and a column is a decision centre. Decision centres are related by decision links (wide arrow) as shown in Figures C.2 and C.5. Besides of decision links, simple information exchanges between decision centres are represented by information links (narrow arrow) as shown in figure C.5.

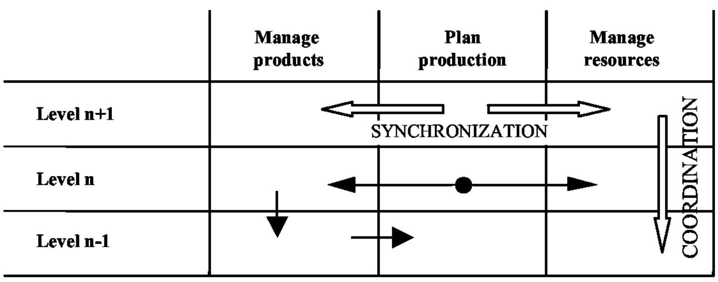


Figure C.5 — Decision view formalism

C.4 Decision View guidance

Based on the practical experiences of applying GRAI methodology ([1], [2]), the following guidance summarize good practice:

a) a decision view model should contain at least three decision levels, namely long-term, medium-term and short-term;

b) decision view model should contain at least three functions; for manufacturing enterprises these are "plan production", "manage resources" and "manage products;"

NOTE 1 Other functions relevant to a study may be added depending on the specificity of the studied company. In some cases, the function “manage resources” makes a distinction between human and machine types of resources; the function “manage products” may also be subdivided into “purchasing” and “inventory”.

c) a given decision level, the horizon **H** should be longer than the activity cycle **Cy** governed by the decision at that level;

NOTE 2 For example, if the decision is concerned with shop floor planning, the horizon should be longer than the manufacturing cycle observed in the workshop.

d) a given decision level, the horizon **H** should be longer than twice the period **P**;

e) levels are classified by decreasing horizons and decreasing periods for equal horizons; and

f) the horizon of level **n** should be at least twice as long as the period of level **n-1**.

Figure C.6 summarizes the time consistency relationships between decision levels when constructing the view.

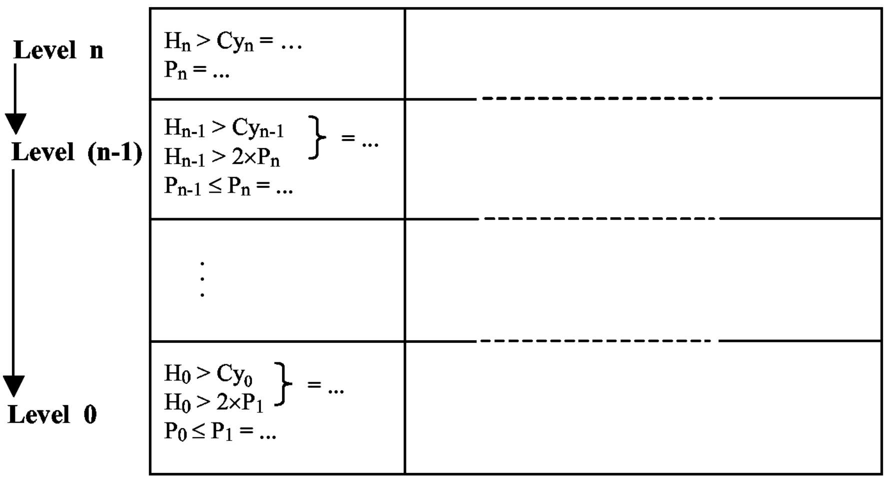


Figure C.6 — Time consistency between decision levels

C.5 Decision View for production planning and control — an example

To achieve a consistent, integrated, system-wide decision-making view, decisions taken within each decision centre should be constrained by a decision frame. The structure and details of these decision frames are not the concern of this annex because they are enterprise-specific. Consequently, as a generic example, the model shown in C.1 does not contain any decision and information links as well as specific values of time horizon and period, because these depend on the specificities of a particular enterprise and will be determined when study a particular enterprise.

NOTE 1 When elaborating the decision view of a particular system, one of the two styles of structure shown in Figure C.2 may be chosen and adapted to define possible decision links between decision centres.

A decision view for integrated production planning and control is shown in Table C.1. Various production planning and control activities are categorized and mapped onto the basic decision system structure.

Table C.1 — Decision view for production planning and control – an example

	Manage products (P ∩ T)	Plan production (P ∩ R ∩ T)	Manage resources (R ∩ T)
Long-term (Strategic in Scope) (H,P)	- define procurement policies (frequency, quantity, etc.) - define inventory policies - determine desired inventory levels - issue purchase request for critical part/material	- determine planning parameters (lot sizing, rules, etc.) - establish long-term production plan (in terms of product families) - define Master Production Schedule (in terms of finished products and major sub-assemblies)	- estimate needs on resource capacity based on the long-term production plan - define resource management policies (including sub contracting decisions) - issue request for resource acquisition (both human and equipment) - plan rough-cut capacity based on the long-term production plan
Medium-term (Tactical in Scope) (H,P)	- implement procurement and inventory policies - issue material (parts) requirements request - adjust long-term decisions if necessary	- establish production plan for manufacturing parts	- implement resource management policies (such as sub contracting) - install new purchased equipment - training new personnel - detail capacity - adjust resource capacity among various sites, shops, cells, etc.
Short-term (Operational in Scope) (H,P)	- monitor reception of purchased material/part - manage material shortage - issue urgent and exceptional material purchase request - make reservation of material/parts to manufacture end-products - manage inventory levels - report on inventory to production scheduling	- define a detailed shop floor production schedule - dispatch manufacturing orders - monitor production in progress - report production status - adjust production schedule to account for availability of resources and shortage of material	- assign personnel to machines based on detailed shop production schedule - manage machine breakdown - deal with absenteeism of human operators

The decision view contains three functional categories:

a) "Manage products" is concerned with the decisions for managing products (raw material, parts or sub-assemblies) to obtain finished goods;

b) “Plan production” is concerned with decisions for the transformation of products by resources. Its main purpose is to manage the production by synchronizing decisions taken by “manage products” and “manage resources;”

c) “Manage resources” is concerned with the decisions for determining resource policies (human or equipment) and managing resource capacity with respect to production loads.

The decision view contains three time categories:

a) Decisions taken at a long-term level deal with production, procurement and resource management policies (e.g. desired inventory level, critical material list, critical resource list, sub contracting or not) and define production objectives to meet for a strategic horizon;

b) Decisions taken at a medium-term level are concerned with the tactical implementation of means necessary to meet the strategic objectives (e.g. procure raw materials; implement supplementary resources, both human and machine; if production volume increases, select sub contractors and define sub contracting items);

c) Decisions taken at short-term level are concerned with the finite capacity scheduling of manufacturing operations, using the tactical means to meet the strategic objectives. These decisions should be made in such a manner that the correct product is manufactured on the correct machine, by the correct person and at the correct time.

NOTE 2 The values of horizon and period vary by level depending on the size and activity of each particular enterprise. For a big company and in the case of a complex product such as an aeroplane, the long-term horizon may be one or two years, while for a small company and a simple product, e.g. furniture, the long-term horizon may be between six months to one year. These values should be determined in the particular context.

C.6 Glossary of references for Decision View

[1] Doumeingts, G. (1984), Méthode GRAI: méthode de conception des systèmes en productique. Automatic Control, University Bordeaux I, 519 p.. (in French)

[2] Doumeingts, G., Vallespir, B. and Chen, D. (1998), Decision modelling GRAI grid, in: Handbook on Architecture for Information Systems (Peter Bernus, Kai Mertins, Gunter Schmidt. (Eds.)), Springer.

[3] CEN TS 14818 (2004): Enterprise Integration – Decisional Reference Model, Technical Specifications, CEN, April 2004.

Annex D (renumbered from existing Annex B), subclause D.6

Add the following reference at the end of the subclause:

"Yuliu Chen and M.M.Tseng. A Stair-Like CIM System Architecture. IEEE Trans. on CPMT Part C, April 1997, pp:101-110."

ICS 25.040.01

Price based on 17 pages

INTERNATIONAL STANDARD

ISO 15704

First edition
2000-06-01

Industrial automation systems — Requirements for enterprise-reference architectures and methodologies

Systèmes d'automatisation industrielle — Prescriptions pour architectures de référence entreprise et méthodologies

Reference number
ISO 15704:2000(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this International Standard ISO 15704 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

International Standard ISO 15704 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 5, *Architecture, communications and integration frameworks*. In preparing this document, substantive contributions were received from groups involved with enterprise-reference architectures such as the Purdue Enterprise-Reference Architecture (PERA), the Graphes et Résultats et Activités Interreliés GRAI Integrated Methodology (GRAI GIM), the Computer Integrated Manufacturing Open System Architecture (CIMOSA), and the Generalised Enterprise-Reference Architecture and Methodology (GERAM).

Annexes A and B of this International Standard are for information only. Annex A is based on version 1.6.2 of GERAM developed by the IFIP/IFAC Task Force on Architectures for Enterprise Integration who granted permission for its inclusion in ISO 15704.

0 Introduction

0.1 Rationale for enterprise-reference architectures and methodologies

Industrial enterprises create and modify manufacturing and business operations to improve performance in local and global markets. In the operational phase, they deploy a variety of resources such as people, information systems, and automated machinery. Individually and collectively these resources provide the functional capabilities required to expedite business processes and their constituent activities. The inter-working of resources needs to be organised and targeted to accomplish the mission. This requires suitable business rules and organisational structures to enable the enterprise to provide products and services to its customers in conformance with agreed upon criteria.

Enterprises operate under uncertain market and environmental conditions so that enterprise engineering may need to be ongoing. It follows that enterprise personnel have a variety of roles to play in the conception and ongoing development of the mission, business rules, business processes, organisational structures, and supporting resources and services. Because of the high levels of complexity involved in enterprise engineering, invariably it is necessary to deploy means of assessing, structuring, coordinating and supporting these engineering activities.

Enterprise-reference architectures underpinned by reference methodologies provide generally applicable means of organising and coordinating engineering projects. By adopting, and as required adapting, a reference methodology and architecture, enterprise personnel can cooperate in progressing enterprise-engineering projects, improving the enterprise and utilisation of resources. By adopting a reference methodology, architecture, and a supporting tool set, it becomes practical for personnel to reuse explicit enterprise designs and models to achieve enterprise engineering on an ongoing basis to realise further improvements in enterprise operation.

Therefore, a vital need is an enterprise engineering and integration reference base providing methodologies and supporting technologies that can realistically treat the problem of enterprise integration.
The work of the IFAC/IFIP (International Federation of Automatic Control/ International Federation for Information Processing) Task Force on Architectures for Enterprise Integration and of many other similar organisations around the world have recently focused their work on this problem in hopes of achieving the generic solution needed. Their work has shown that such a reference base can be devised, and must be underpinned by an enterprise-reference architecture that:

a) can model the whole life history of an enterprise-integration project from its initial concept through definition, functional design or specification, detailed design, physical implementation or construction, operation to decommissioning or obsolescence;

b) encompasses the people, processes, and equipment involved in performing, managing, and controlling the enterprise mission.

It is important to note that enterprise-reference architectures deal with the structural arrangement (organisation) of the development and implementation of a project or programme such as an enterprise-integration or other enterprise-development programme. In contrast to these enterprise-reference architectures, system architectures deal with the structural arrangement (design) of a system; for example, the computer-control-system part of an overall enterprise-integration system.

The IFAC/IFIP Task Force on Architectures for Enterprise Integration has developed the definition of a complete, generalised enterprise-reference architecture and methodology and has called it GERAM, described in annex A. GERAM will be used as the example reference for the requirements set forth in this document.

0.2 Key principles of enterprise integration

Several concepts that describe the nature of enterprise-reference architectures and methodologies have emerged from the studies of the IFAC/IFIP Task Force on Architectures for Enterprise Integration that can greatly simplify, integrate, and extend the work of enterprise engineering. This work has led to the development of GERAM, which is capable of supporting those who plan, design, and implement complex enterprise-integration projects.

Key principles of an enterprise-reference architecture are described below to provide a basis for the requirements of clause 4.

0.2.1 Applicability to any enterprise

The early work in CIM (computer-integrated manufacturing) 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 involved 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 there are problems within the mission system, manufacturing or other customer product and service operations, or in the associated human and organisational area whose solution would greatly ease the overall system problem, that is, a total solution must involve information, culture, and mission.

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

0.2.2 Enterprise identification and mission definition

No enterprise can exist in the long term without a business or mission, that is, it must produce products or services desired by its customers. It 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 project.

0.2.3 Separation of mission-fulfillment functions from mission-control functions

There are only two basic classes of functions involved in operating any enterprise. These are described below.

a) 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 would include all material and energy transformation tasks and the movement and storage of materials, energy, goods-in-process, and products; and services.

b) The other class comprises functions involved that manage and control the mission-fulfillment to achieve the desired economic or other gains that assure the viability or continued successful existence of the enterprise. These include the collection, storage, and use (transformations) of information to control the business processes, that is, 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.

0.2.4 Identification of process structures

Enterprise operation consists of many transformations of material, energy, and information that can be categorised into two distinct classes: one for information transformations and the other for material and energy transformations. These transformations will be carried out by many separate activities 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.

0.2.5 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 types in both classes mentioned in 0.2.4. Other processes may be automated or mechanised. There are only three classes of implemented tasks or business processes, which are as follows:

a) information and control activities that can be automated by computers or other control devices;

b) mission activities that can be automated by the mission-fulfillment equipment;

c) activities carried out by humans, whether of the information and control or mission-fulfillment class.

It is desirable to have a simple way of showing where and how the human fits in the enterprise and how the distribution of functions between humans and machines is accomplished.

0.2.6 Recognition of enterprise life-cycle phases

All enterprises, of whatever type, follow a life cycle from their initial concept in the mind of an entrepreneur through a series of stages comprising their development, design, construction, operation and maintenance, refurbishment or obsolescence, and final disposal.

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 could build a manufacturing plant (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 (see 0.2.1).

A particular distinction can be made between those life-cycle phases which are concerned with the creation and modification of enterprise entities (its development, design, construction, etc.) and their use (operation). This distinction enables the orderly move (release) from the engineering environment to the operation environment, providing for validation, testing and release of engineering results prior to operation.

0.2.7 Evolutionary approach to enterprise integration

The integration of all of the informational and customer-product and service functions of an enterprise may be a part of a master plan. The actual implementation of such integration may be broken up into a series of co-ordinated projects that are within the financial, physical, and technical capabilities of the enterprise. These projects can be carried out individually or collectively, as these resources allow, as long as the master plan is followed.

0.2.8 Modularity

Because of the massive nature of all enterprise integration projects, modularity should be enforced whenever possible. Thus it would be helpful if all activities were defined in a modular fashion, along with their required interconnections, so they may 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 optimisation techniques as long as 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, national and/or internationally agreed upon standards, the interchange and substitution noted above will be greatly facilitated.

0.3 Aim and benefits of deploying enterprise-reference architectures and methodologies

An enterprise-reference architecture with its associated methodologie and related enterprise-engineering technologies that fulfill the requirements of this standard will enable an enterprise-integration-planning team to determine and develop a course of action that is complete, accurate, properly oriented to future business developments, and carried out with the minimum of resources, personnel, and capital. That is, to:

a) describe the tasks required;

b) define the necessary quantity of information;

c) specify relationships among humans, processes, and equipment in the integration considered;

d) address management concerns;

e) address relevant economic, cultural, and technological factors;

f) detail the extent of computer-support required;

g) support process-oriented modelling that can model the whole life history of an enterprise.

0.4 Benefits of this standard

The enterprise-reference architecture and methodology requirements in this standard will allow a specific enterprise-reference architecture and methodology to be checked for completeness with respect to its current and future purpose. This standard will help guide their development.

This 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 a terminology that pertains specifically to the company, industry, and culture involved. This standard will help guide that selection or creation.

Industrial automation systems — Requirements for enterprise-reference architectures and methodologies

1 Scope

This International Standard defines the requirements for enterprise-reference architectures and methodologies, as well as the requirements that such architectures and methodologies must satisfy to be considered a complete enterprise reference architecture and methodologies.

The scope of these enterprise-reference architectures and methodologies covers those constituents deemed necessary to carry out all types of enterprise creation projects as well as any incremental change projects required by the enterprise throughout the whole life of the enterprise, including

a) enterprise creation,

b) major enterprise restructuring efforts, and

c) incremental changes affecting only parts of the enterprise-life cycle.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 14258, *Industrial automation systems — Concepts and rules for enterprise models.*

ISO 14258, *Industrial automation systems — Concepts and rules for enterprise models: Technical Corrigendum 1.*

3 Terms and definitions

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

3.1
activity
all or part of functionality

NOTE Enterprise activity consists of elementary tasks performed in the enterprise that consume inputs and allocate time and resources to produce outputs.

3.2
architecture
a description (model) of the basic arrangement and connectivity of parts of a system (either a physical or a conceptual object or entity)

NOTE There are two, and only two, types of architectures that deal with enterprise integration. These are:

a) system architectures (sometimes referred to as "type 1" architectures) that that deal with the design of a system, e.g. the computer control system part of an overall enterprise integration system;

b) enterprise-reference projects (sometimes referred to as "type 2" architectures) that deal with the organisation of the development and implementation of a project such as an enterprise integration or other enterprise development programme.

3.3
attribute
a piece of information stating a property of an entity

NOTE An attribute models an intrinsic property of something, for example, the geometry of a part, the condition of a tool, or the qualifications of a worker.

3.4
behaviour
how the whole or part of the system acts and reacts

3.5
business process
a partially ordered set of enterprise activities that can be executed to realise a given objective of an enterprise or a part of an enterprise to achieve some desired end-result

3.6
enterprise
one or more organisations sharing a definite mission, goals, and objectives to offer an output such as a product or service

NOTE This term includes related concepts such as extended enterprise or virtual enterprise.

3.7
enterprise engineering
the discipline applied in carrying out any efforts to establish, modify, or reorganise any enterprise

3.8
enterprise model
a representation of what an enterprise intends to accomplish and how it operates

NOTE An enterprise model, which is used to improve the effectiveness and efficiency of the enterprise, identifies the basic elements and their decomposition to any necessary degree. It also specifies the information, resources and organisational requirements of these elements, and provides the information needed to define the requirements for integrated information systems.

3.9
framework
a structural diagram that relates the component parts of a conceptual entity to each other

NOTE Neither the structure involved nor the relationship of the parts to each other have a life cycle or time relationship in contrast to the enterprise-reference ("type 2") architecture.

3.10
genericity
the extent to which a concept is generic

3.11
life cycle
the finite set of generic phases and steps a system may go through over its entire life history

3.12
life history
the actual sequence of steps a system has gone through during its lifetime

3.13
master plan
the documentation of the major engineering and operations planning effort carried out prior to any large enterprise integration or other systems engineering project

NOTE The master plan is based on management goals for the project and uses functional and economic analysis techniques for the preliminary engineering of the project to achieve an initial design specification and prove economic feasibility.

3. 14
methodology
a set of instructions (provided through text, computer programs, tools, etc.) that is a step-by-step aid to the user

NOTE In carrying out needed aspects of the life cycle of the entity integration project, the methodology prescribes or describes the processes of enterprise engineering and integration. A methodology may take account of any involved social, political and economic aspects.

3.15
mission
that activity in which an enterprise engages to fulfil the customer product or service function for which it was established; the mechanism by which an enterprise achieves its goals and objectives

3.16
model
an abstract representation of reality in any form (including mathematical, physical, symbolic, graphical, or descriptive form) to present a certain aspect of that reality for answering the questions studied

NOTE A model can be used to describe the enterprise activities or the different phases of the life cycle of the enterprise (see 3.8).

3.17
organisation
the structure of an enterprise and the distribution of responsibilities and authorities in the enterprise

3.18
resource
an enterprise entity that provides some or all of the capabilities required by the execution of an enterprise activity and/or business process

3.19
structure
the definition of the relationships among the components of an organization

3.20
system
a collection of real-world items organised for a given purpose

NOTE A system is characterised by its structure and its behaviour.

4 Requirements for enterprise-reference architectures and methodologies

4.1 Applicability and coverage of enterprise-entity types

4.1.1 Generality

Enterprise-reference architectures and methodologies shall be capable of assisting and structuring the description, development, operation, and organisation of any conceivable enterprise entity, system, organisation, product, process, and their supporting technology. There may be reference architectures that cover a sub-set and therefore are confined to a specific class or type of enterprise or systems (such as discrete parts manufacturing, process industries, and information systems). However, the area covered by these reference architectures and methodologies shall be clearly identified.

The methodology associated with a reference architecture shall provide the necessary guidelines and management techniques for the initiation and pursuit of a project or program of development and operation of an enterprise or entity. Such a methodology may or may not be model-based. That is, the enterprise engineering process may or may not result in a specific enterprise model.

Enterprise-reference architectures and methodologies need not be based on any one single methodology and its accompanying architecture or framework. There are potentially many different methodologies and/or frameworks that might be used for it. The primary consideration shall be applicability and capability in relation to these requirements.

Enterprise-reference architectures and methodologies shall identify concepts and components as described in 4.2 and 4.3.

4.1.2 Enterprise design

Enterprise-reference architectures and methodologies shall identify the activities needed to manage, conceive/define, describe, design, implement, maintain, and decommission any enterprise entity. See 3.2.3 and 3.4 of ISO 14258.

4.1.3 Enterprise operation

Enterprise-reference architectures and methodologies shall identify the activities needed to use the results of enterprise engineering in the operation itself. Such use may include model-based decision support and model-driven operation monitoring and control.

4.2 Concepts

4.2.1 General

The enterprise-reference architectures and methodologies shall address the role of humans, the description of processes (function and behaviour) and the representation of all supporting technologies throughout the life cycle of the enterprise.

4.2.2 Human oriented

Enterprise-reference architectures and methodologies shall exhibit the capability to represent human aspects, such as organisational and operational roles, capabilities, skills, know-how, competencies, responsibilities, authorisation, and relations to the organisation.

4.2.3 Process oriented

Enterprise-reference architectures and methodologies shall exhibit the capability to represent the enterprise operation. Such representations shall cover both the functionality and behaviour of the operation. The representations shall recognise the life cycle and life-history concepts of enterprise-entity types and shall support process-oriented operations.

4.2.4 Technology oriented

Enterprise-reference architectures and methodologies shall exhibit the capability of representing all technologies employed in the enterprise operation.

NOTE Such representation of 4.2.2, 4.2.3, and 4.2.4 shall provide for integration-technology infrastructures used to support enterprise engineering and operation of business processes, models of enterprise resource (information technology, manufacturing technology, office automation and others), facility layout models, information-system models, communication-system models and logistics models.

4.2.5 Mission-fulfillment oriented

Enterprise-reference architectures and methodologies shall exhibit the capability to represent any process and its constituent activities involved in performing the established mission of the enterprise in terms of providing the enterprise products and services to its customers.

4.2.6 Mission-control oriented

Enterprise-reference architectures and methodologies shall exhibit the capability to represent any process and its constituent activities of the accomplishment of the management and control of the established mission of the enterprise according to the criteria established by enterprise management.

4.2.7 Framework for enterprise modeling

Enterprise-reference architectures and methodologies that are model-based shall exhibit the capability to model entities within the conceptual space defined by the dimensions of life cycle, genericity, and modelling views.

NOTE These dimensions are discussed further in ISO 14258.

4.2.8 Life cycle

Enterprise-reference architectures and methodologies shall identify and represent the life-cycle phases that are pertinent during the life of any enterprise entity.

NOTE Life-cycle phases encompass all activities from inception to decommissioning (or end of life) of the enterprise entity which might be characterised. There is no presumption that these phases are necessarily sequential.

4.2.9 Life history

An enterprise-reference architecture and methodology shall be capable of representing the life history of any enterprise entity; that is, the representation in time of activities carried out on any enterprise entity.

NOTE Using the life-cycle concept of 4.2.8, the user can identify these activities as life-cycle-activity types while the life history allows the same user to identify the corresponding time element. This demonstrates the iterative nature of the life-cycle concept compared with the time sequence of life history. These iterations identify different change processes required on the operational processes and/or the product or customer services.

4.2.10 Modelling views

The enterprise-reference architectures and methodologies that are model-based shall provide concepts for representing different views (see 3.7 of ISO 14258) of an enterprise model to allow it to be described as an integrated model but to be presented to the user in different subsets. Views contain a subset of facts present in the integrated model in order to concentrate on relevant questions that the respective stakeholders may wish to consider using enterprise modelling. Different views may be made available highlighting certain aspects of the model and hiding others. The concept of view is applicable to models of all entity types across their entire life cycle.

Enterprise-reference architectures and methodologies that are model-based shall include these four model-content views: function, information, resource, and organisation.

4.2.11 Genericity

Those reference architectures and methodologies that are model-based shall provide the capability for representing generic-enterprise elements (4.3.3), partial-enterprise models (4.3.4), and particular-enterprise models (4.3.5).

4.3 Components of enterprise-reference architectures

4.3.1 Engineering methodologies

Enterprise-reference architectures and methodologies shall provide enterprise-engineering methodologies that guide the user in the process of management of change and provide methods of progression for every type of life-cycle activity for any enterprise-entity type.

Enterprise-engineering methodologies shall describe the process of enterprise integration and 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.

4.3.2 Modelling languages

Enterprise-reference architectures and methodologies that are model-based shall identify enterprise modelling languages or modelling constructs that allow the enterprise operation to be described. See 3.2 and 3.6 of ISO 14258.

Modelling constructs shall allow users to represent the different elements of the modelled-enterprise entity and thereby improve both modelling efficiency and model understanding. The form (representation) of modelling constructs shall be adapted to the needs of people creating and using enterprise models. Therefore, different languages may exist to accommodate the aspects of different users (e.g. business users, system designers, information-technology modelling specialists, and others). In addition, modelling languages may allow the formation of higher level constructs out of more basic constructs (macro constructs) to enhance modelling productivity.

Enterprise 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 can be described in terms of ontological theories. This is especially important if enterprise models are to support the enterprise operation itself, because such models must be executable. However, the definition of the formal semantics shall be supported by natural language explanations of the concepts as well.

4.3.3 Generic Elements

Enterprise-reference architectures and methodologies can be based on generic elements of enterprise design and modelling. Such generic elements are, in increasing order of formality, glossaries, meta-models, and ontological theories. These elements provide for consistency of enterprise representations.

4.3.4 Partial models

Enterprise-reference architectures and methodologies that are model-based shall support the concept of partial-enterprise models (reusable reference models). This allows the user to capture and reuse concepts common to many enterprises and thereby to increase modelling efficiency. Partial models still need adaptation to the requirements of the specific enterprise. Partial models can cover one or all concepts identified in 4.2.

4.3.5 Particular models

Enterprise-reference architectures and methodologies that are model-based shall support the creation of particular-enterprise models that describe part or all of any enterprise entity.

Enterprise models may be expressed in enterprise modelling languages and may be maintained, created, analysed, stored, and distributed using enterprise-engineering tools. Both model creation and model use may be supported by integrating-information-technology services. The use of such services will ensure access to real-time information in the engineering and operational environments of the enterprise.

4.3.6 Tools

Enterprise-reference architectures and methodologies shall be supported by computer based tools that aid the user in enterprise engineering and integration projects. Such tools shall be based on one or more enterprise-engineering methodologies and may have implemented one or more modelling language.

Such tools shall provide analysis and simulation capabilities for the creation, manipulation, use, and management of enterprise designs and models, as well as their analysis, description, and evaluation. These functions are needed for decision making in the course of enterprise engineering. In addition, such tools may support collaborative work across organisation boundaries.

Engineering tools can enable the user to connect enterprise designs and models with the real business process, so as to keep the designs and models up-to-date.

4.3.7 Modules

Enterprise-reference architectures and methodologies shall provide the capability for representing the concept of enterprise modules or implemented building blocks or systems (products, or families of products) that are utilised as common resources in enterprise engineering and enterprise integration. One set of enterprise modules important to enterprise engineering and integration is the integrating infrastructure or the set of integration-technology services required for enterprise engineering and operation in heterogeneous environments.

4.3.8 Enterprise-operational systems

One result of the enterprise-engineering process shall be a design or model for the enterprise-operational system. The enterprise operational system shall consist of the hardware and software needed to fulfil the enterprise objectives and goals. The content of the operating system is derived from enterprise requirements.

4.4 Representation

Enterprise-reference architectures and methodologies shall provide a mechanism to guide users in the use of the associated components of 4.3, e.g. a framework or high-level, graphical interpretation. The framework or graphical form should show the applicability of, and relations between, the different components.

4.5 Glossary

To promote understanding about projects and other co-operative efforts, enterprise-reference architectures and methodologies shall provide a

a) consistent glossary and a semantics and syntax for use in enterprise-engineering and integration efforts, or

b) reference to other suitable glossaries.

5 Completeness and compliance

The degree of completeness of a candidate architecture and methodology shall be determined by their limits of applicability as defined by 4.1 and the extent to which they employ the concepts and components of 4.2 and 4.3. Consequently the degree of completeness from this point of view shall be measured by any restrictions to a particular class or type of enterprise.

Any assessment of the degree of compliance of a candidate architecture and methodology shall be qualified by the following:

a) a preliminary statement as to whether or not they are model based;

b) a statement of the degree to which they then conform partially or totally to the appropriate requirements of 4.2 to 4.5.

In the event of partial compliance, areas of non-conformance shall be explicitly identified.

Annex A
(informative)

GERAM: Generalised Enterprise-Reference Architecture and Methodologies

A.1 Introduction

A.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 and be reactive so that change and adaptation should be a natural dynamic state rather then something occasionally forced onto the enterprise. This necessitates the integration of the enterprise operation[1] and the development of a discipline that organises 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 Engineering[2].

Previous research, carried out by the AMICE Consortium on CIMOSA, by the GRAI Laboratory on GRAI and GIM, and by the Purdue Consortium on PERA, (as well as similar methodologies by others) has produced reference architectures which were meant to be organising all enterprise integration knowledge and serve as a guide in enterprise engineering programs. The IFIP/IFAC Task Force 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 generalised architecture is the outcome of the work of the Task Force.

Starting from the evaluation of existing enterprise integration architectures (CIMOSA, GRAI/GIM and PERA), the IFAC/IFIP Task Force on Architectures for Enterprise Integration has developed an overall definition of a generalised architecture. The proposed framework was entitled 'GERAM' (Generalised Enterprise Reference Architecture and Methodology). GERAM is about those methods, models and tools which 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).

GERAM, as presented below, defines a tool-kit of concepts for designing and maintaining enterprises for their entire life-history. GERAM is not yet-another-proposal for an enterprise reference architecture, but is meant to organise existing enterprise engineering knowledge. 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.

A.1.2 Scope

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

- major enterprise engineering/enterprise integration efforts (green field installation, complete re-engineering, merger, reorganisation, formation of virtual enterprise or consortium, value chain or supply chain integration, etc.);
- incremental changes of various sorts for continuous improvement and adaptation.

---

[1] Enterprise integration is about breaking down organisational barriers and improving interoperability to create synergy within the enterprise to operate more efficiently and adaptively.

[2] Enterprise Engineering is the collection of those tools and methods which one can use to design and continually maintain an integrated state of the enterprise.

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

An important aspect of enterprise engineering is the recognition and identification of feedback loops on various levels of enterprise performance as they relate to its products, mission and meaning. 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 on process and organisation are required. The continuous use of these feedback loops will be the prerequisite for the continuous improvement process of the enterprise operation and its adaptation to the changes in the relevant market, technology and society.

A.2 The framework for enterprise engineering and enterprise integration

A.2.1 General

GERAM provides a description of all the elements recommended in enterprise engineering and integration and thereby sets the standard for the collection of tools and methods from which any enterprise would benefit to more successfully tackle initial integration design, and the change processes which may occur during the enterprise operational lifetime. It does not impose any particular set of tools or methods, but defines the criteria to be satisfied by any set of selected tools and methods. GERAM views enterprise models as an essential component of enterprise engineering and integration; this includes various formal (and less formal) forms of design descriptions utilised 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 A.1 and is briefly described in this clause. Each component is then defined in more detail in A.3.

The GERAM framework identifies in its most important component GERA (Generalised Enterprise Reference Architecture) the basic concepts to be used in enterprise engineering and integration (for example, enterprise entities, life cycles and life histories of enterprise entities). GERAM distinguishes between the methodologies for enterprise engineering (EEMs) and the modelling languages (EMLs) which are used by the methodologies to describe and model, the structure, content and behaviour of the enterprise entities in question. These languages will enable the modelling of the human part in the enterprise operation as well as the parts of business processes and their supporting technologies. The modelling process produces enterprise models (EMs) which represent all or part of the enterprise operations, including its manufacturing or service tasks, its organisation and management, and its control and information systems. These models can be used to guide the implementation of the operational system of the enterprise (EOSs) as well as to improve the ability of the enterprise to evaluate operational or organisational alternatives (for example, by simulation), and thereby enhance its current and future performance.

The methodology and the languages used for enterprise modelling are supported by enterprise engineering tools (EETs). The semantics of the modelling languages may be defined by ontologies, meta models and glossaries which are collectively called generic enterprise modelling concepts (GEMCs). The modelling process is enhanced by using partial models (PEMs) which are reusable models of human roles, processes and technologies.

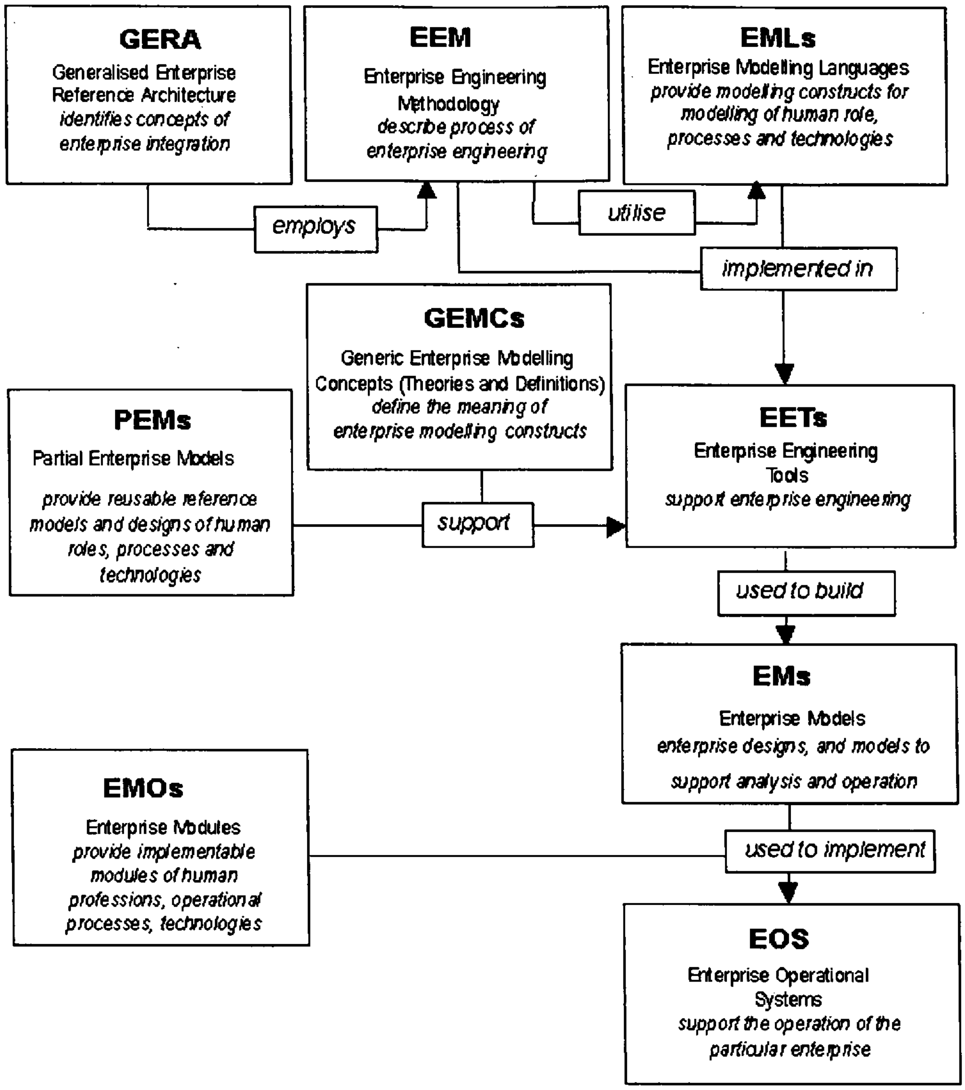


Figure A.1 — GERAM (Generalised Enterprise Reference Architecture and Methodology) framework components

The operational use of enterprise models is supported by specific modules (EMOs) which provide prefabricated products like human skill profiles for specific professions, common business procedures (e.g. banking and tax rules) or IT infrastructure services, or any other product which can be used as a component in the implementation of the operational system (EOSs).

Potentially, all proposed reference architectures and methodologies could be characterised in GERAM so that developers of particular architectures could gain from being able to commonly refer to the capabilities of their architectures, without having to rewrite their documents to comply with GERAM. Users of these architectures would benefit from GERAM because the GERAM definitions would allow them to identify what they could (and what they could not) expect from any chosen particular architecture in connection with an enterprise engineering methodology and its proposed supporting components.

A.2.2 Definition of GERAM Framework Components

A.2.2.1 GERA – Generic Enterprise Reference Architecture

GERA defines the enterprise related generic concepts recommended for use in enterprise engineering and integration projects. These concepts can be categorised as:

a) human oriented concepts

   1) to describe the role of humans as an integral part of the organisation and operation of an enterprise, and

   2) to support humans during enterprise design, construction and change;

b) process oriented concepts for the description of the business processes of the enterprise;

c) technology oriented concepts for the description of the business process supporting technology involved in both enterprise operation and enterprise engineering efforts (modelling and model use support).

A.2.2.2 EEMs – Enterprise engineering methodologies

EEMs describe the processes of enterprise engineering and integration. An enterprise engineering methodology may be expressed in the form of a process model or structured procedure with detailed instructions for each enterprise engineering and integration activity.

A.2.2.3 EMLs – Enterprise modelling languages

EMLs define the generic modelling constructs for enterprise modelling adapted to the needs of people creating and using enterprise models. In particular enterprise modelling languages will provide construct to describe and model human roles, operational processes and their functional contents as well as the supporting information, office and production technologies.

A.2.2.4 GEMCs – Generic enterprise modelling concepts

GEMCs define and formalise the most generic concepts of enterprise modelling. Generic Enterprise modelling concepts may be defined in various ways. In increasing order of formality generic enterprise modelling concepts may be defined as:

- natural language explanation of the meaning of modelling concepts (glossaries);
- some form of meta-model (e.g. entity relationship meta schema) describing the relationship among modelling concepts available in enterprise modelling languages;
- ontological theories defining the meaning (semantics) of enterprise modelling languages, to improve the analytic capability of engineering tools and, through them, the usefulness of enterprise models (typically, these theories would be built inside the engineering tools).

A.2.2.5 PEMs – Partial enterprise models

PEMs (reusable-, paradigmatic-, typical models) capture characteristics common to many enterprises within or across one or more industrial sectors. Thereby these models capitalise on previous knowledge by allowing model libraries to be developed and reused 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 humans roles (skills and competencies of humans in enterprise operation and management), operational processes (functionality and behaviour) and technology components (service or manufacturing oriented), infrastructure components (information technology, energy, services, etc.).

Partial models may cover the whole or a part of a typical enterprise. They may concern various enterprise entities such as products, projects, companies, and may represent these from various points of view such as data models, process models, organisation models, to name a few.

Partial enterprise models are also referred to in the literature as 'Reference Models', or 'Type 1 Reference Architectures'[3]. These terms have the same meaning.

A.2.2.6 EETs – Enterprise engineering tools

EETs support the processes of enterprise engineering and integration by implementing an enterprise engineering methodology and supporting modelling languages. Engineering tools should provide for analysis, design and use of enterprise models.

A.2.2.7 EMs – (Particular) enterprise models

EMs represent the particular enterprise. Enterprise models can be expressed using enterprise modelling languages. EMs include various designs, models prepared for analysis, executable models to support the operation of the enterprise, etc. They may consist of several models describing various aspects (or views) of the enterprise.

A.2.2.8 EMOs – Enterprise modules

EMOs are products which can be utilised in the implementation of the enterprise. Examples of enterprise modules are human resources with given skill profiles (specific professions), types of manufacturing resources, common business equipment or IT infrastructure (software and hardware) intended to support the operational use of enterprise models.

Special emphasis is on the IT infrastructure which will support enterprise operations as well as enterprise engineering. The services of the IT infrastructure will provide two main functions:

a) model portability and interoperability by providing an integrating infrastructure across heterogeneous enterprise environments;

b) model driven operational support (decision support and operation monitoring and control) by providing real-time access to the enterprise environment.

The latter functionality will be especially helpful in the engineering tasks of model update and modification. Access to real world data provides much more realistic scenarios than for model validation and verification than simulation based on 'artificial' data.

A.2.2.9 EOSs – (Particular) enterprise operational systems

EOSs support the operation of a particular enterprise. Their implementation is guided by the particular enterprise model which provides the system specifications and identifies the enterprise modules used in the implementation of the particular enterprise system.

A.3 Description of GERAM Framework Components

A.3.1 GERA – Generalised Enterprise Reference Architecture

A.3.1.1 General

GERA defines the generic concepts recommended for use in enterprise engineering and integration projects. These concepts can be classified as follows.

---

3 Life-cycle architectures such as GERA are referred to as 'Reference Architectures of type 2'.

a) Human oriented concepts, covering human aspects such as capabilities, skills, know-how and competencies as well as roles of humans in the enterprise organisation and operation. The organisation related aspects have to do with decision level, responsibilities and authorities, the operational ones relate to the capabilities and qualities of humans as enterprise resource elements. In addition, the communication aspects of humans have to be recognised to cover interoperation with other humans and with technology elements when realising enterprise operations. Modelling constructs will be required to facilitate the description of human roles as an integral part of the organisation and operation of an enterprise. The constructs should facilitate the capture of enterprise models which describe

   1) human roles,

   2) the way in which human roles are organised so that they interoperate with other human and technology elements when realising enterprise operations, and

   3) the capabilities and qualities of humans as enterprise resource elements.

   An appropriate methodology will also be required which promotes the retention and reuse of models that encapsulate knowledge (i.e. know-how possessed by humans expressed as an enterprise asset) during the various life phases of enterprise engineering projects.

b) Process oriented concepts, dealing with enterprise operations (functionality and behaviour) and covering enterprise entity life cycle and activities in various life-cycle phases, life history, enterprise entity types, enterprise modelling with integrated model representation and model views.

c) Technology oriented concepts, dealing with various infrastructures used to support processes and include for instance resource models (information technology, manufacturing technology, office automation and others), facility layout models, information system models, communication system models and logistics models.

Examples of enterprise reference architectures are provided by ARIS[4], CIMOSA[5], GRAI/GIM[6], IEM[7], PERA[8]. ENV 40003 defines a general framework for enterprise modelling. ISO 14258 defines rules and concepts for enterprise models.

A.3.1.2 Human oriented concepts

The role of humans in the enterprise remains fundamental. However sophisticated and integrated an enterprise can be, humans will always make the final decision. With the emergence of decentralised organisations, 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 will prove to be very useful and enable flexible reaction to environmental changes. In addition the different factors describing the capabilities of humans have to be captured as well. Human factors are concerned with professional skills, experience, etc.

---

4 ARIS: Architektur für Informations Systeme (Architecture for Information Systems)

5 CIMOSA: CIM Open Systems Architecture

6 GRAI/GIM: Graphe et Résultats et Activités Interreliés (Graphs with Results and Activities Interrelated)/ GRAI Integrated Methodology

7 IEM: Integrated Enterprise Modelling

8 PERA: Purdue Enterprise Reference Architecture

Typically humans may assume different roles during enterprise engineering and operation. Examples are: chief executive, chairperson, marketing, sales, technical (R&D), 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. Also alternative organisational structures may be deployed, for example elements of an organisation may be linked hierarchically or heterarchically and demonstrate properties of holons, webs, nets, temples or clusters. Further organisational structuring may occur on a functional, process or geographic basis.

Often humans and groupings of humans will be assigned a number of roles and responsibilities which need to be carried out concurrently and cohesively, where each may 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 resources effectively and collectively under complex and changing circumstances is key to the competitive position of an enterprise.

Although it is not practical to model all aspects of human roles within an enterprise, concepts are required to formally represent those human factors connected with enterprise integration. This should be achieved in a way that harmonises human roles with that of other human and technology elements, as an integral part of the organisation and operation of an enterprise. Hence the need for constructs which promote the capture of knowledge (possessed by humans) in the form of reusable enterprise models about

- the role of individuals and groups of individuals,
- the way in which organisational structures and constraints are applied to co-ordinate those roles, such as via the delegation of responsibilities and control and reporting procedures, and
- the capabilities and qualities of humans, treated as resource elements.

It is important to understand when, by whom and how decisions are made in the enterprise as well as who can fulfil certain tasks in the replacement of others.

Knowledge about the roles of humans and ways in which those roles can be harmonised can be capitalised and reused as an enterprise asset. The degree to which such knowledge can be formalised within computer processable models will directly influence the degree to which it can be capitalised. Computer processable models naturally facilitate analysis, transformation, storage and integration (based on common understandings). Whereas mental models retained and processed by humans will be less tractable for such purposes. However the retention and reuse of informal models (such as in the form of cause and effect relationships and shared mental models or images) can also be of significant benefit in realising improved cohesion in an enterprise. Hence even where formal modelling of human issues proves impractical the retention and reuse of knowledge should be encouraged by deploying suitable social processes, human organisational structures and methodologies and tools which promote explicit model capture and visualisation.

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 utilisation of resources as new products and services are launched, and
- improve its resilience to the loss of core competancies in the form of knowledgeable human assets.

A taxonomy of human factors and their relation to the activity model would allow to relate human aspects to enterprise models. Needed are human role models on decision making, capabilities, socio-technical models (motivation, incentives, etc.), skill models, and organisational models, with others to be determined.

Human role models will support the definition of human responsibilities and authorisation in both the enterprise operation and its organisation. Such models will support the collection of relevant information and its recognition in the design of the operational system. GERA caters for human factors in its view concept (see A.3.1.5.3). This concept provides in its process oriented model views and technology oriented implementation view for the recognition of humans and the capturing of relevant information. Also the role of humans as an operational resource is recognised in these views. In this role the human skills and capabilities will be described. The human aspect of enterprise integration must also be thoroughly dealt with in the change methodology both in the human's role of change agent and in the role of potential and actual resource (see A.3.2).

A.3.1.3 Process oriented concepts

A.3.1.3.1 General

Business process-oriented modelling aims at describing the processes in the enterprise capturing both their functionality (that is what has to be done and by what role) and their behaviour (that is when things are done and in which sequence). In order to achieve a complete description of the processes a number of concepts have to be recognised in the guiding methodology. The process-oriented concepts defined in GERA are:

- enterprise entity life cycle and life-cycle phases;
- life history,
- enterprise entity types, and
- enterprise modelling with integrated model representation and model views.

These concepts will be described in more detail below.

A.3.1.3.2 Life cycle

A.3.1.3.2.1 General

Figure A.2 shows the GERA life cycle for any enterprise or any of its entities. The different life-cycle phases define types of activities that are pertinent during the life of the entity. Life-cycle activities encompass all activities from identification to decommissioning (or end of life) of the enterprise or entity. A total of seven life-cycle activity types have been defined, which my be subdivided further as demonstrated for the design type activities which have been broken down into two lower level types of activities (based on the customary subdivision in many industries of design into preliminary- and detailed design activities). The life-cycle diagram used in the description of the life cycle of an entity is itself a model of the enterprise engineering methodology.

A.3.1.3.2.2 Entity identification

This is the set of activities which identify the contents of the particular entity under consideration in terms of its boundaries and its relation to its internal and external environments. These activities include the identification of the existence and nature of a need (or need for change) for the particular entity. In other words these are the activities which define what is the entity of which the life cycle is being considered.

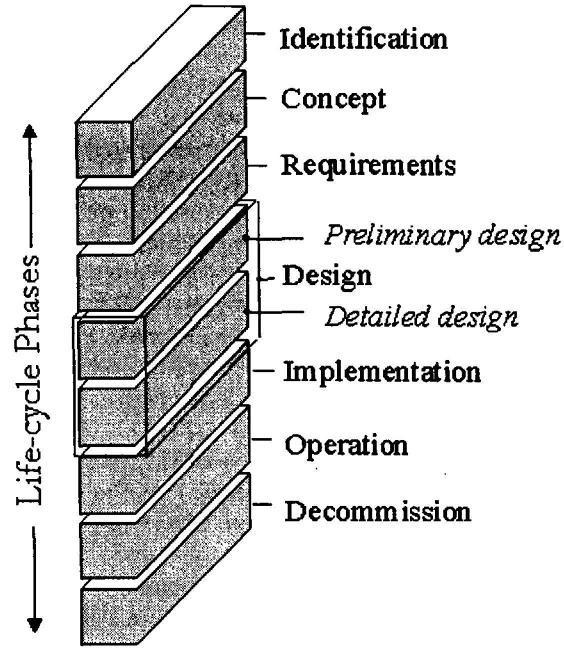


Figure A.2 – GERA life-cycle phases for any enterprise or entity

A.3.1.3.2.3 Entity concept

The set of activities that are needed to develop the concepts of the underlying entity. These concepts include the definition of the entity's mission, vision, values, strategies, objectives, operational concepts, policies, business plans and so forth.

A.3.1.3.2.4 Entity requirement

The activities needed to develop descriptions of operational requirements of the enterprise entity, its relevant processes and the collection of all their functional, behavioural, informational and capability needs. This description includes both service and manufacturing requirements and management and control requirements of the entity – no matter whether these will be satisfied by humans (individuals or organisational entities), or machinery (including manufacturing-, information-, control-, communication-, or any other technology).

A.3.1.3.2.5 Entity design

The activities which support the specification of the entity with all of its components that satisfy the entity requirements. The scope of design activities includes the design of all human tasks (tasks of individuals and of organisational entities), and all machine tasks concerned with the entity's customer services and products and the related management and control functions. The design of the operational processes includes the identification of the necessary information and resources (including manufacturing-, information-, communication-, control- or any other technology).

Any life-cycle stage may be subdivided into sub-stages to provide additional structuring of life cycle activities. For example, see figure A.2 in which the design activity is divided into functional design and specification and detailed design to permit the separation and classification of:

a) overall enterprise specifications (sufficient to obtain approximate costs and management approval of the ongoing project);

b) major design work necessary for the complete system design suitable for fabrication of the final physical system.[9]

A.3.1.3.2.6 Entity implementation

The activities which define all those tasks which must be carried out to build or re-build (i.e. manifest) the entity. This comprises implementation in the broadest sense, covering:

a) commissioning, purchasing, (re)configuring or developing all service, manufacturing and control software as well as hardware resources;

b) hiring and training personnel, and developing or changing the human organisation;

c) component testing and validation, system integration, validation and testing, and releasing into operation.

Note that the implementation description (documentation) may deviate from the design specification of the entity due to preferences or unavailability of specified components.

A.3.1.3.2.7 Entity operation

The activities of the entity which are needed during its operation for producing the customers product or service which is its special mission along with all those tasks needed for monitoring, controlling, and evaluating the operation. Thus the resources of the entity are managed and controlled so as to carry out the processes necessary for the entity to fulfil its mission. Deviations from goals and objectives or any feedback from the environment may lead to requests for change, which includes enterprise re-engineering or continuous improvement of its human and technology resources, its business processes, and its organisation.

A.3.1.3.2.8 Entity decommissioning

These activities are needed for re-missioning, retraining, redesign, recycling, preservation, transfer, disbanding, disassembly, or disposal of all or part of the entity at the end of its useful life in operation.

A.3.1.3.3 Life history

The life history of a business entity is the representation in time of life-cycle tasks carried out on the particular entity during its entire life span. Relating to the concept described above, the concept of life history allows to identify the tasks pertaining to these different phases as activity types. This demonstrates the iterative nature of the life-cycle concept compared with the time sequence of life history. These iterations identify different change processes required on the operational processes and, or the product or customer services.

---

[9] Note that a) the need for such subdivision is found methodologically very important (see the Purdue guide *A Handbook on Master Planning and Implementation for Enterprise Integration Programs*), and b) the wording allows for the consistency of this life-cycle phase definition with the ENV 40 003 which has only one design phase. The reason for this difference is that the Purdue guide considers PERA and thus GERA as the model of the methodology, and in that case the subdivision is essential. On the other hand CIMOSA and thus ENV 40 003 considers the life-cycle phases to be characterisations of modelling levels or languages. From this latter aspect the subdivision is not necessarily essential, because the preliminary and detailed design differ only in design detail. Using this wording GERA can play both roles a and b.

Typically, multiple change processes are in effect at any one time, and all of these may run parallel with the operation of the entity. Moreover, change processes may interact with one another. Within one process, such as a continuous improvement project, multiple life-cycle activities would be active at any one time. For example, concurrent engineering design and implementation processes may be executed 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 are made up of processes which in turn rely on the same type of life-cycle activities as defined in the GERA life cycles (A.3.1.3.2). For this reason life-cycle activities are a useful abstraction in understanding the life history of any entity.

Figure A.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, and one decommissioning process.

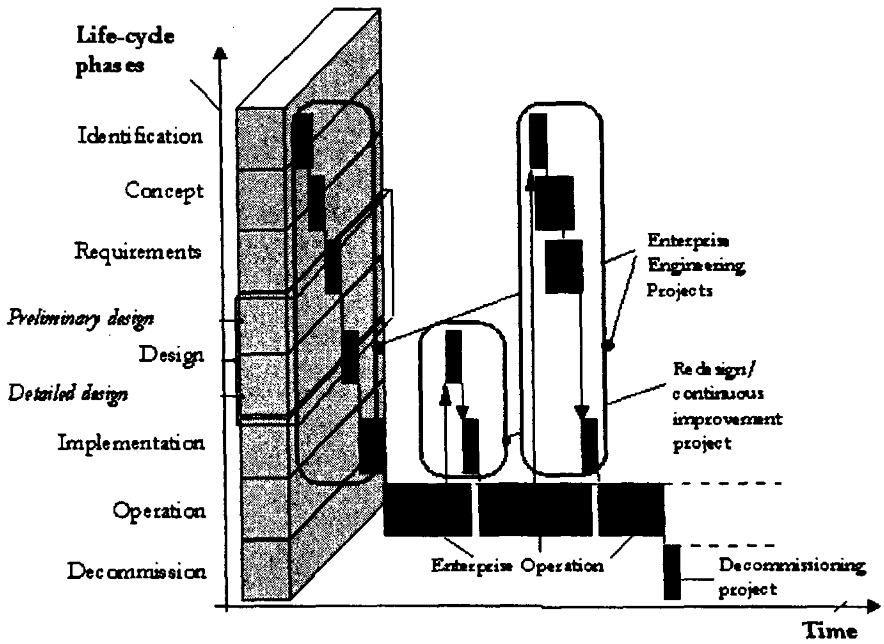


Figure A.3 – Parallel processes in the entity's life-history

A.3.1.3.4 Entity types in enterprise integration

A.3.1.3.4.1 General

Figure A.4 shows how the life-cycle activities of two entities may be related to each other. The operation of entity A supports the life-cycle activities for design and implementation of entity B. For example, entity A may be an engineering entity producing part of entity B, such as a factory.

Conversely the life-cycle activities of entity A need to be supported with information about the life-cycle details of entity B. That is, to identify a plant, to define its concepts and requirements, and to design it, one must use information about which life-cycle activities of the plant's products need to be covered in the operation of this plant.

Examples of other relations between the life-cycle activities of enterprise entities may be defined. However, it is always the case that only the operational activities of entities influence the life-cycle activities of other activities. GERA introduces the concept of entity types and the relations between the different types. Many categories of enterprise entities can be defined. In the following two different ways of categorising enterprise types will be discussed: an operation oriented set and a generic and recursive set of enterprise entity types. The two sets have close relations to each other and both identify the product entity as the result of the operation of other entities.

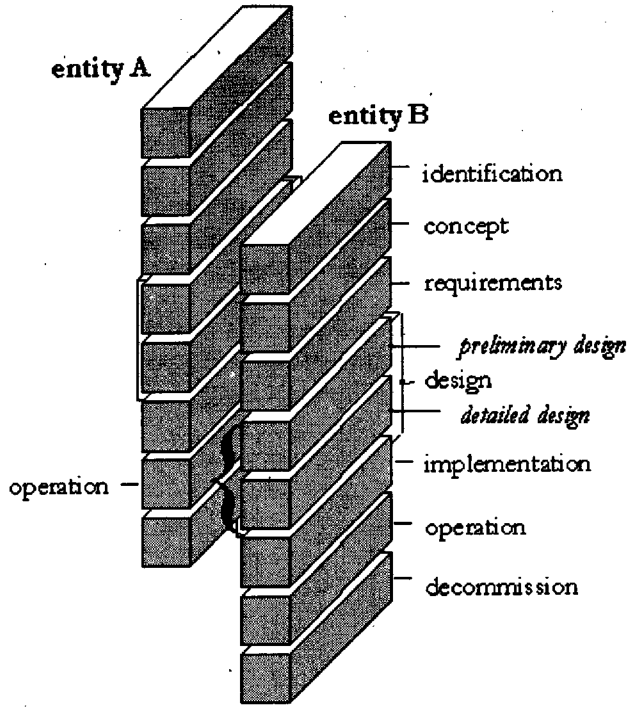


Figure A.4 – Example for the relationship between life cycles of two entities

A.3.1.3.4.2 Operation oriented enterprise entity types

These enterprise entity types, described below, are all concerned with different types of operations.

A.3.1.3.4.2.1 Project Enterprise Entity (Type A)

This type defines an enterprise (often with a short life history) which is created for the one-off production of another entity. Examples of project enterprise are: enterprise engineering project, one of a kind manufacturing projects, building projects, etc.

The project enterprise is characterised by its close linkage with the life cycle of the single product or service which it is producing. The management system of project enterprises is typically set up quickly, while the rest is created and operated 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. A typical example would be 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).

A.3.1.3.4.2.2 Repetitive – Service and Manufacturing – Enterprise Entity (Type B)

This type defines enterprises supporting a type- or a family of products, produced in a repetitive or sustained mode. During their life history these business enterprises undergo multiple change processes. Examples of repetitive business enterprise are service enterprises, manufacturing plants, engineering firms, infrastructure enterprises, etc.

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

A.3.1.3.4.2.3 Product Entity (Type C)

This type defines 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 their life cycles are described by GERAM.

A.3.1.3.4.3 Recursive enterprise entity types

A generic and recursive set of four enterprise entity types which have been defined as follows.

- Strategic Enterprise Management Entity (Entity Type 1), which defines the necessity and the starting of any enterprise engineering/integration effort.
- Enterprise Engineering/Integration Entity (Entity Type 2), which provides the means to carry out the enterprise engineering efforts defined by enterprise Entity Type 1. It employs a methodology (Entity Type 5) to define, design, implement and build the operation of the enterprise entity (Entity Type 3).
- Enterprise Entity (Entity Type 3), which is the result of the operation of Entity Type 2. It uses a methodology (Entity Type 5) and the operational system provided by Entity Type 2 to define, design, implement and build the products and customer services of the enterprise (Entity Type 4).
- Product Entity (Entity Type 4), which is the result of the operation of Entity Type 3. It represents all products and customer services of the enterprise.

This set may be complemented by a fifth entity type that represents the methodology needed for guiding the enterprise engineering and enterprise integration activities.

- Methodology Entity (Entity Type 5), which represents the methodology to be employed by any enterprise entity type in the course of its operation, which operation in general leads to the creation of another entity type.

The recursiveness of the first four entity types can be demonstrated by identifying the role of the different entities, their 'products' and the relations between them. Figure A.5 represents the chain of enterprise entity developments. The Entity Type 1 will always start creation of any lower level entity by identifying goal, scope and objectives for the particular entity. Development and implementation of a new enterprise entity (or new business unit) will then be done by a Entity Type 2 whereas a Entity Type 3 will be responsible for developing and manufacturing a new product (Entity Type 4). With the possible exception of the Entity Type 1 all enterprise entities will have an associated entity life cycle. However, it is always the operational part of the entity life cycle in which the lower entity is defined, created, developed and built. The operation itself is supported by an associated methodology for enterprise engineering, enterprise operation, product development and production support.

Figure A.5 shows both the life cycle of the methodology (Entity Type 5) and the process model of the methodology. There must be a clear distinction between the life cycle of the methodology (which is essentially the description of how a methodology is developed) and its process model which is the individual manifestation of the methodology entity itself used to support the operational phase of particular enterprise entities.

The operational relations of the different entity types are also shown in Figure A.6 which demonstrates as an example the contributions of the different entities to the life cycle of a manufacturing entity (Entity Type 3). The manufacturing entity itself produces the enterprise product (Entity Type 4) in the course of its operation phase.

The defined set of entity types is seen to be sufficient to allow representation of other types as well. For example, the distinction between on-of-a-kind or project related enterprise entities and continuous operation type enterprises would only require different parts of the life-cycle activities to be used in the life history of such entities. This is already indicated in Figure A.3 in which the engineering processes could relate to an Entity Type 2 and the operational processes to an Entity Type 3 which produces the product or customer services (Entity Type 4). The involvement of Entity Type 1 depends on the degree of change to be carried out in the change process.

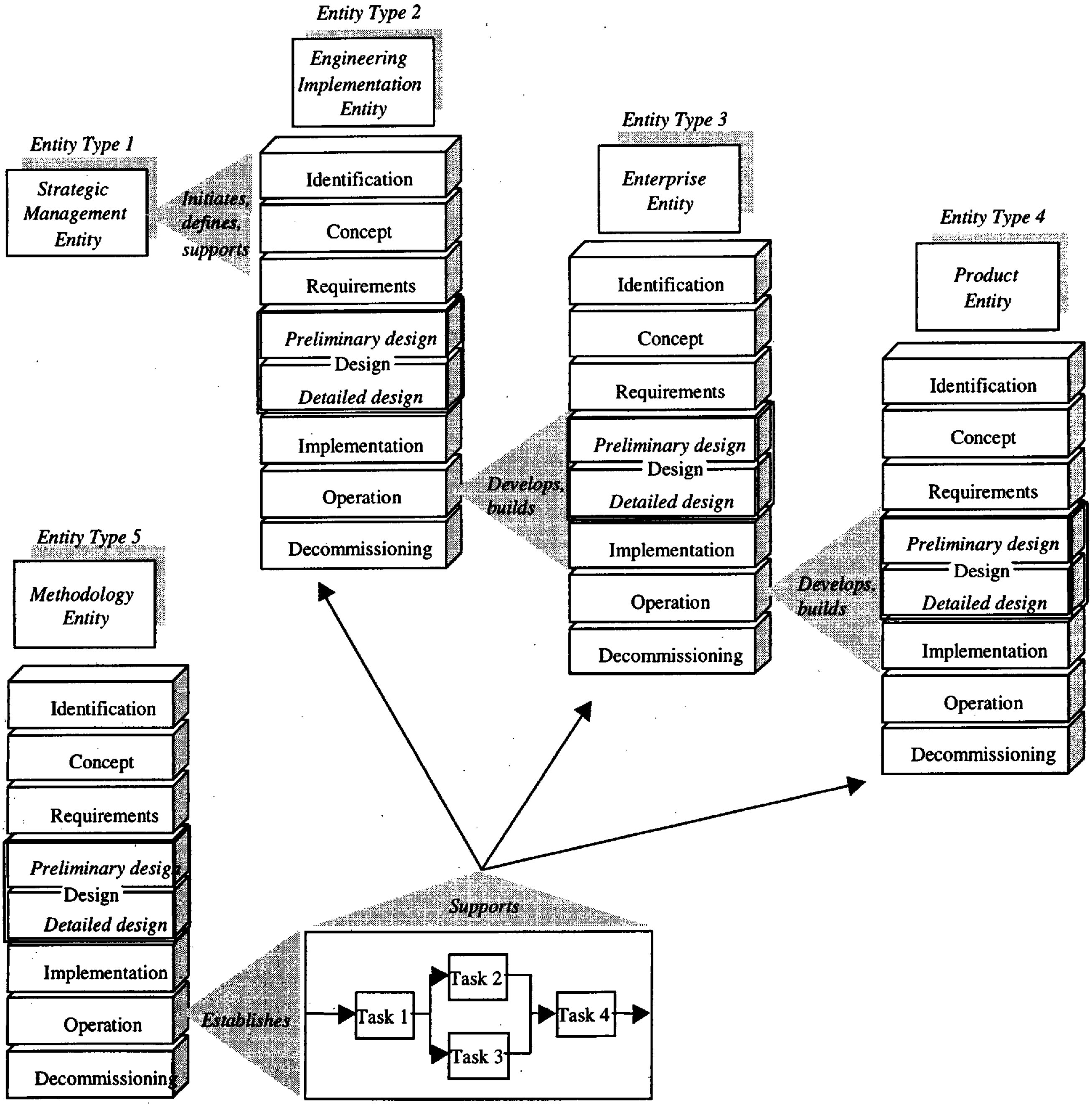


Figure A.5 – Relationships between life cycles of GERA entity types

A.3.1.3.5 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, organisation, products etc. of the enterprise. Process modelling allows the user to represent the operation of enterprise entities and entity types in all their aspects: functional, behaviour, information, resources and organisation. This provides for operational use of the models for decision support by evaluating operational alternatives and for model driven operation control and monitoring.

A.3.1.4 Technology oriented concepts

A.3.1.4.1 General

Both the enterprise engineering process and the operational environment employ 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 and information processing and information sharing. Technology-oriented concepts have to provide descriptions of the technology involved in both the enterprise operation and the enterprise-engineering efforts.

For the operation oriented technology such concepts have to relate to models such as resource models and resource organisation models (e.g. shop floor models, system architectures, information models, infrastructure models), communication models (e.g. network models), etc.

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

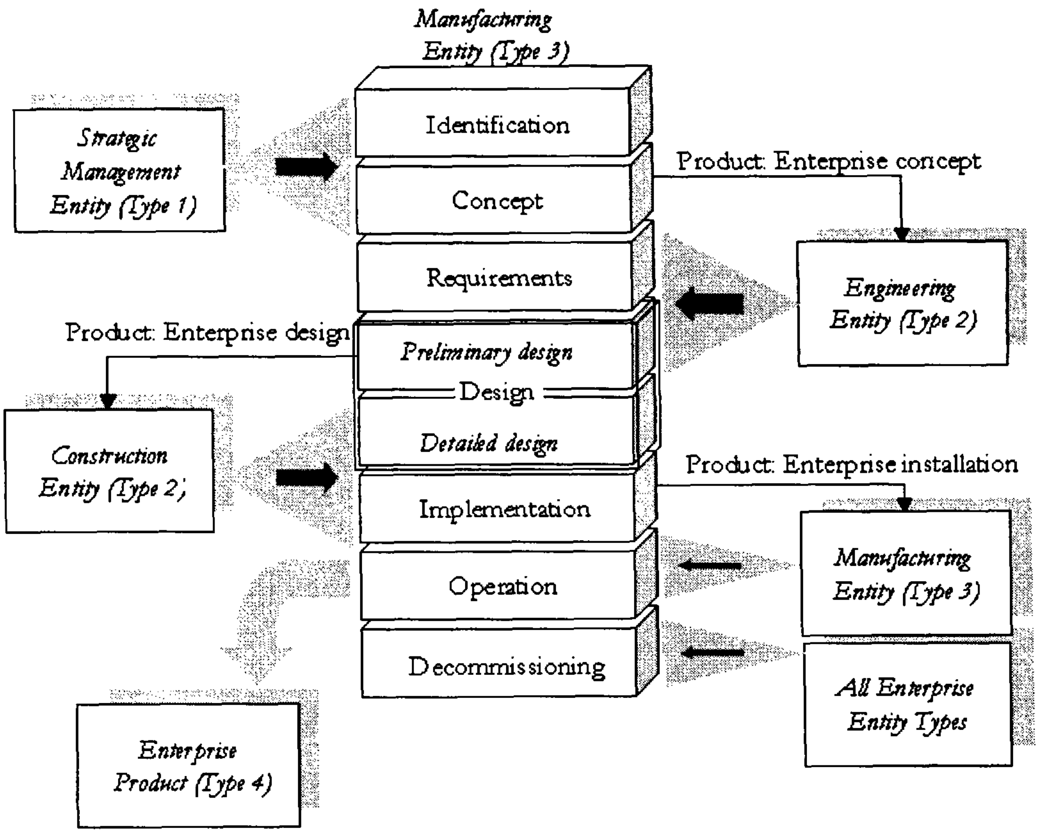


Figure A.6 – Relationships between GERA entity types

A.3.1.4.2 IT support for enterprise engineering and enterprise integration

IT support for enterprise engineering as well as enterprise operation should provide two main functions:

a) model portability and interoperability by providing an integrating infrastructure across heterogeneous enterprise environments;

b) model driven operational support (decision support and operation monitoring and control) by providing real time access to the enterprise environment.

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

A.3.1.4.3 Enterprise Model Execution and Integration Services (EMEIS)

To illustrate the potential use of computer executable models for on-line operation of the enterprise, Figure A.7 illustrates the concept of an integrating infrastructure linking the enterprise model to the real world systems. Integrating services act as a harmonising 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 harmonising characteristics of the integrating infrastructure enables transfer of information across and beyond the organisation. Through the semantic unification of the modelling framework interoperability of enterprise models is assured as well.

Efforts aimed at enterprise modelling support have been implemented in pilot implementations (CCE/CNMA, CIM-BIOSIS, CIMOSA, MIDA, OPAL, PISA, TOVE). Some of these project results have been evaluated in a CEN/TC 310 report and have lead to statement of requirements for enterprise model execution and integration services by CEN/TC 310 as well. The statement of requirements distinguishes between the model development services (MDS), the model execution services (MXS) and the general IT services (see Figure A.7). However no explicit service entities have been defined.

Relevant standardisation is in progress at the European level (see work item *"CIM Systems Architecture - Enterprise Model Execution and Integration Services"* CEN/TC 310/WG1, 1994).

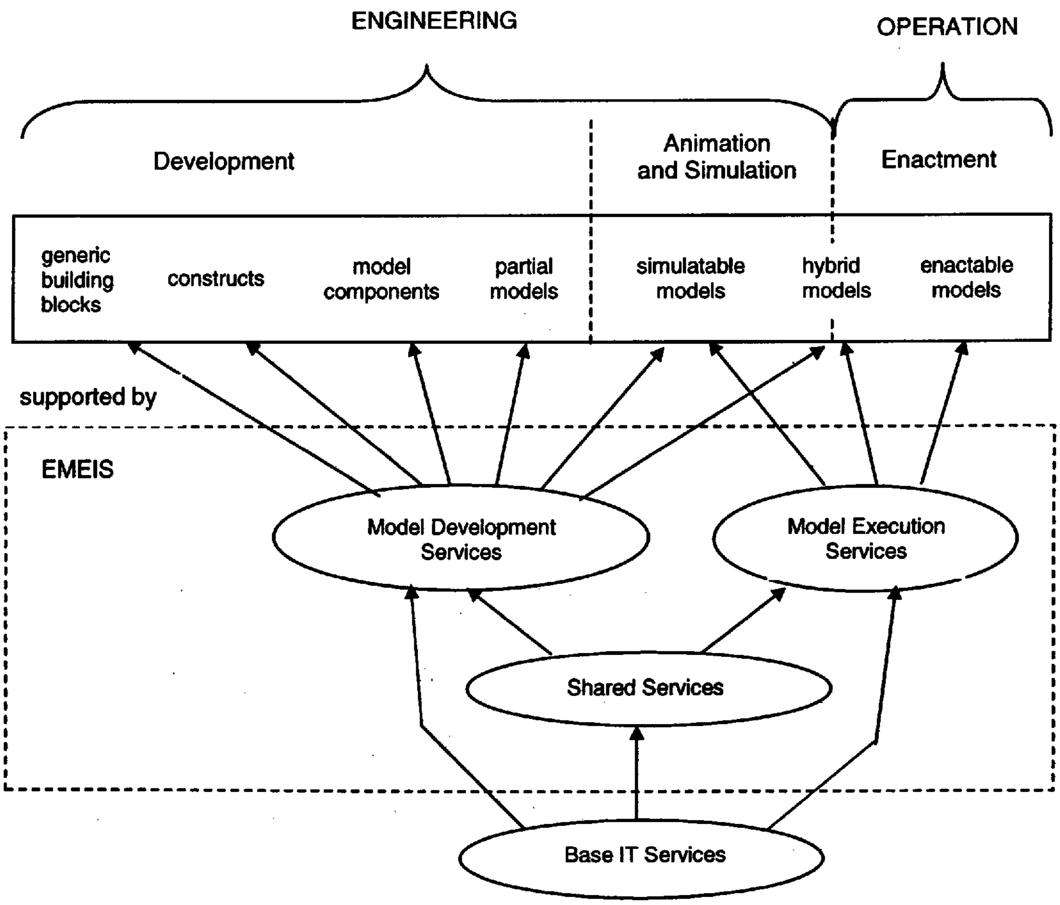


Figure A.7 – Reference model of EMEIS

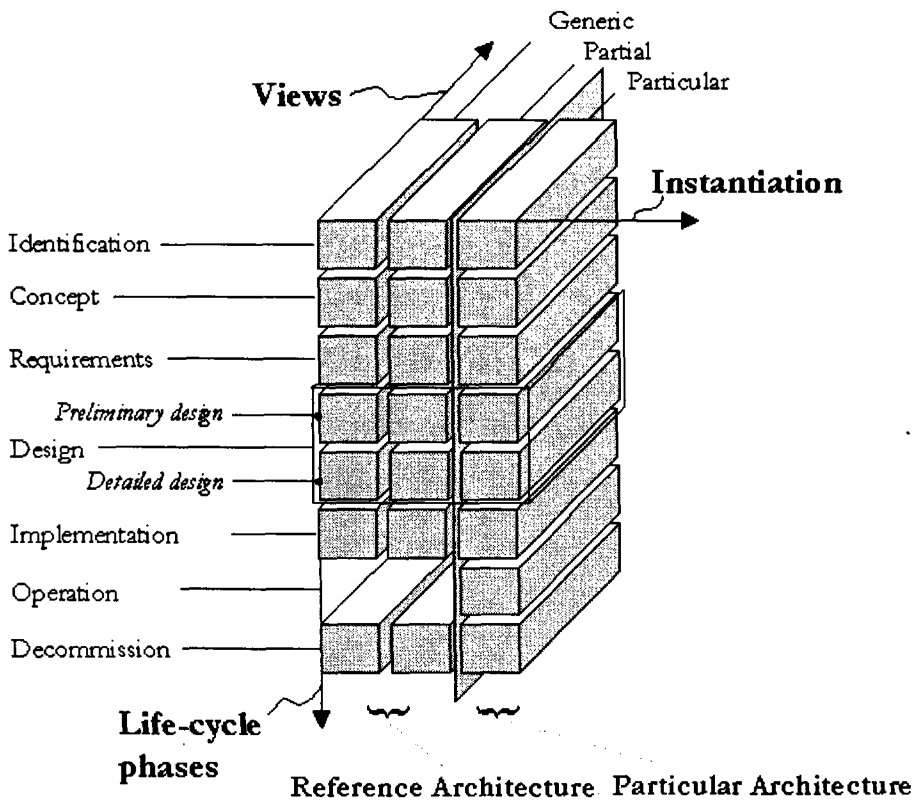
A.3.1.5 Modelling framework of GERA

A.3.1.5.1 General

GERA provides an analysis and modelling framework which is based on the life-cycle concept and identifies these 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 the life-cycle activities;
- Genericity Dimension, providing for the controlled particularisation (instantiation) process from generic and partial to particular;
- View Dimension, providing for the controlled visualisation of specific views of the enterprise entity.

Figure A.8 shows the three dimensional structure identified above which represents this modelling framework. The reference part of the modelling framework itself consists of the generic and the partial levels only. These two levels organise into a structure the definitions of concepts, basic and macro level constructs (the modelling languages), defined and utilised for the description of the given area. The particular level 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 particular set of life-cycle activities. However, it is intended that the modelling languages should support the two-way relationship between models of adjacent life-cycle phases. That is, the derivation of models from an upper to a lower state or the abstraction of lower models to an upper state, rather than having to create different models for the different sets of life-cycle activities.



NOTE The left hand side represents the reference models, the right hand side represents the resulting particular enterprise models.

Figure A.8 – The GERA modelling framework

A.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, organisation, products etc. of the enterprise). The life-cycle activities of an entity may define various models of that entity to be created. That is, the results of enterprise modelling are all the various designs, models prepared for analysis, executable models to support enterprise operation, and so on (see reference [13] of A.4.1).

The emphasis in enterprise modelling is currently on process and product models for representing *enterprise operations.* Process oriented modelling allows representing the operation of enterprise entities and entity types in all their aspects: functional, behaviour, information, resources and organisation. This provides for operational use of the models for decision support by evaluating operational alternatives and for model driven operation control and monitoring.

Enterprise models in general represent a very complex reality. In order to reduce this complexity enterprise models have to allow the representation of certain aspects (views) of the model. Aspects which represent part of the model which is relevant to the concerns of the user. This allows the manipulation of the model according to the user's concerns, without being disturbed by the overall complexity of an overall total 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 (such as engineering drawings, charts etc.) also belong to this category. The reason for the emphasis on process modelling is only because this is a relatively new activity in enterprise design not earlier practised. This modelling activity is, however, over and above the already practised ones, not to replace them.

A.3.1.5.3 View concepts

A.3.1.5.3.1 General

To decrease the apparent complexity of the resulting enterprise models GERA provides the view concept which allows the operational processes to be described as an integrated model, but to be presented to the user in different sub-sets (model views) of an integrated model (see Figure A.9). Views contain a subset of facts present in the integrated model allowing the user to concentrate on relevant questions that the respective stakeholders may wish to consider using enterprise modelling. Different views may be made available highlighting certain aspects of the model and hiding all others. The concept of view is applicable for models of all entity types across their entire life cycle. Modelling views are generated from the underlying integrated model. Any model manipulation (any change of the contents of a particular view), will be reflected in all relevant views and aspects of the model.

GERA defines a "finest mesh of subdivision" of the kinds of models deemed desirable, allowing for the fact that an even finer subdivision may be prescribed by a GERAM-compliant candidate architecture. The following subdivisions of models or model views have been identified in GERA:

- Entity Model Content Views: function, information, resource, organisation;
- Entity Purpose Views: customer service and product, management and control;
- Entity Implementation View: human implemented tasks, automated tasks (management and control technology and mission-support technology);
- Entity Physical Manifestation Views: software, hardware.

Additional views may be defined according to specific user needs.

GERAM does not require every view to be present in every life-cycle phase. However, it requires that the scope of the defined views is covered by any other different view subdivision. Thereby it is guaranteed that all relevant facts are captured. For example, it is not as important to have a separate software view and a separate hardware view as it is to model both software and hardware. The Enterprise Engineering Methodology decides which model to produce 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 concepts such as:

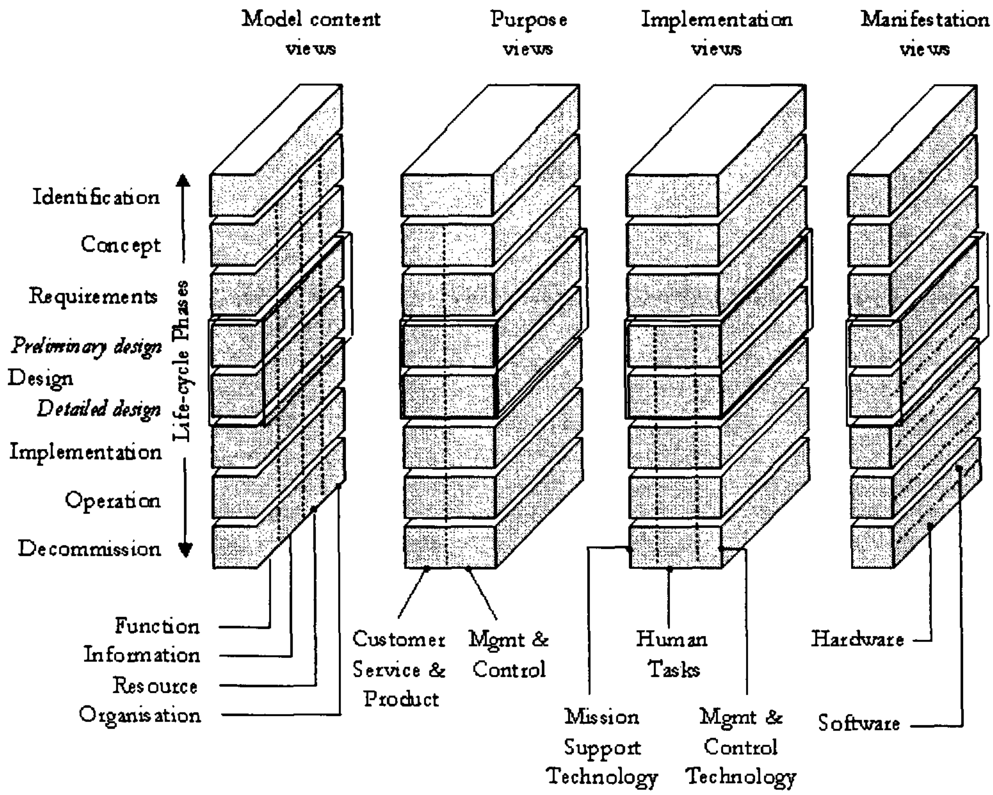
a) a design choice;

b) to simulate a process to find some process characteristics, such as cost or duration;

c) to analyse an existing process for finding inconsistencies or other problems in the information or material flow;

d) to analyse decision functions and find missing decisional roles.

The view concept is the generalisation of the view concepts of many architectures including CIMOSA, GRAI (and others). The GERA modelling framework allows for languages of different *expressive power* for each model view. This means that there is a choice of language in any particular view depending on what analysis capability (and therefore expressive power) is required, according to the enterprise engineering methodology's needs.



NOTE Four essential view types and their contents are shown. Other modelling views may be defined if needed and supported by the engineering tools.

Figure A.9 – The Modelling View concept

A.3.1.5.3.2 Entity model content views

Four different model content views have been defined for the user-oriented, process representation of the enterprise entity descriptions: Function, Information, Organisation, and Resource.

The Function View represents the functionalities (activities) and the behaviour (flow of control) of the business processes of the enterprise. Decisional activities of the management related operations are represented, 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 that the management- and control system of the enterprise is often called the decision system.) The function view includes functional models, process models, decision models, which differ in their expressive power (and competency, e.g. in terms of what analysis questions these models can answer) but all talk about some aspect of the enterprise function. As a result, the "function view" is a holding place for a host of possible models. Examples include CIMOSA (see references [8] and [14] of A.4.1) function view models, GRAI Grid (see reference [15] of A.4.1) and GRAI Net representations of decision centres, Petri nets, Event Driven Process Chains, Generalised Process Networks, QGERT, and GPSS models. All of the above types of models belong to the "function" view. Similar arguments can be developed for the information, for the organisation, and for the resource view.

The Information View collects the knowledge about objects of the enterprise (material and information) as they are used and produced in the course of the enterprise operations. The information is identified from the relevant activities and is structured into an enterprise information model in the information view for information management and the control of the material and information flow.

The Resource View represents the resources (humans and technical agents as well as technological components) of the enterprise as they are used in the course of the enterprise operations. Resources will be assigned to activities according to their capabilities and will be structured into resource models, e.g. for asset management.

The Organisation View represents the responsibilities and authorities on all entities identified in the other views (processes, information, and resources). It caters for the structure of the enterprise organisation by organising the identified organisational units into larger units such as departments, divisions, sections, etc.
Other modelling views may be defined if needed (such as ecological, economic) and supported by the engineering tools.

The entity model content views in particular covers a great deal. This is because there are many different languages which fit any given model view in this category.

A.3.1.5.3.3 Entity purpose views

Two different views, described below, allow to represent the model contents according to the purpose of the enterprise entity.

- The Customer Service and Product View represents the contents relevant to the enterprise entity's operation and to the operation results. This represents the mission of the enterprise entity being studied.

- The Management and Control View represents the contents relevant to management and control functions necessary to control that part of the enterprise entity which produces products or delivers services for the customer.

This view subdivision is defined to delineate the scope of the description of the enterprise, maintaining that the scope should extend to both the mission fulfilment part and the management part of the enterprise. An enterprise engineering methodology may propose that separate models or descriptions be prepared for these two parts.

A.3.1.5.3.4 Entity implementation views

The implementation of the enterprise entity may be presented in two different views, described below, based on the division between human and automated tasks.

- The Human Activities Views represents all information related to the tasks to be done by humans. The view distinguishes between the tasks that may be done by humans (extent of humanizability) and those that must be done by humans (extent of automation).

- The Automated Activities View presents all the tasks to be done by machines. This includes information related to those tasks to be carried out by mission support technology and those carried out by management and control technology (i.e. 'technology tasks'). The implementation view distinguishes between the tasks that may be done by machines (extent of automatability) and those that must be done by machines (extent of automation).

A.3.1.5.3.5 Entity physical manifestation views

Two different views, described below, allow the user to represent the physical manifestation of the enterprise entity.

- The Software View represents all information resources capable of controlling the execution of the operational tasks in the enterprise. Examples are any computer program which can be stored in a computer or in any other control device enabling the execution of an operation task, or any set of instructions for humans with defined skills such that the instructions are for the humans to perform a task which they otherwise would not have been able to carry out. Software is also a controllable state, e.g. a configuration description of manufacturing hardware, such that the hardware in that configuration can perform a task provided the configuration is maintained for the duration of that task.

- The Hardware View represents all physical resources which have the capability to perform some sets of tasks in the enterprise. Examples are a computer system with given performance characteristics, an employee with given skills, or a machine with given functionality.

Figure A.10 shows an overlay of the different views identified above. The view categories are in general independent of each other, but certain combinations may be useful to represent specific aspects of the enterprise at particular life cycle phases. The availability of any view is subject of its implementation in the supporting engineering tool.

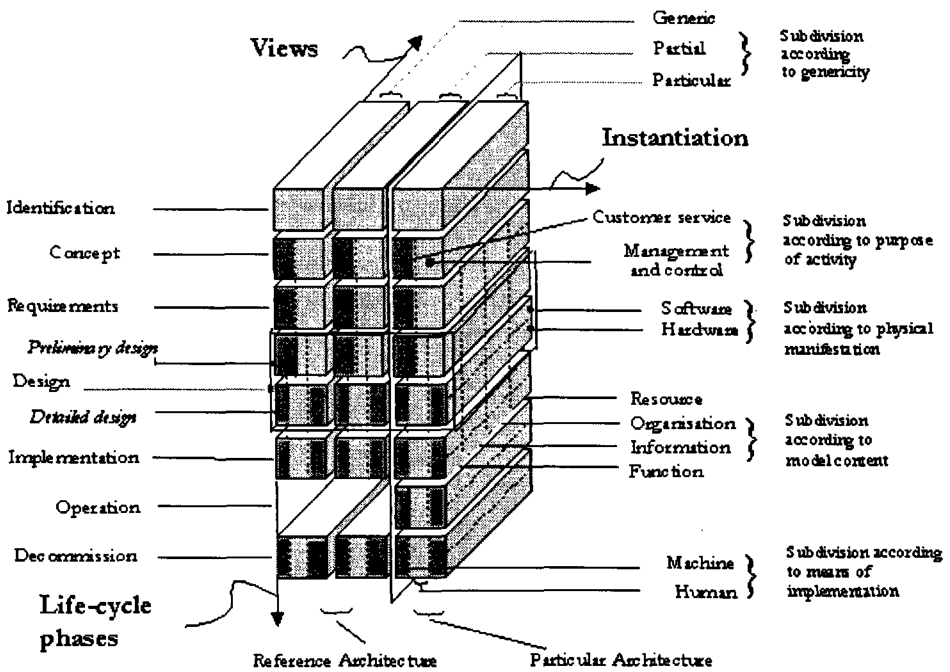


Figure A.10 – GERA modelling framework with modelling views

A.3.2 EEMs – Enterprise engineering methodologies

A.3.2.1 General

Enterprise engineering methodologies describe the processes of enterprise engineering. Their scope is defined in the GERA life-cycle concept. A generalised methodology like generalised architectures is applicable to any enterprise regardless of the industry involved.

An enterprise engineering methodology will help the user in the process of the enterprise engineering of integration projects whether the overall integration of a new or revitalised enterprise or in management of on-going change. It provides methods of progression for every type of life-cycle activity. The upper two sets of these activities (identification and concept) are partly management and partly engineering analysis and description (modelling) tasks. The requirements and design activity sets are mostly oriented towards engineering tasks throughout the process, including the production of enterprise models and designs throughout the process.

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

The enterprise engineering 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 engineering may itself be carried out as a specific project. But the integration task may start at any one of the enterprise's life-cycles activities, not necessarily in the top 'identification' ones. For example: a given engineering project of a new plant may not have to start with the identification and concept definition of the plant, because the customer (who commissioned the design and building of the plant) may have already carried out these activities. In this case the engineering project enterprise should only specify the requirements and carry out the design/detailed design, and implementation (building) of the plant. Such engineering project will then use the requirements, design, and implementation parts of a complete enterprise engineering methodology.

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 particular engineering task.

Enterprise engineering methodologies may be described in terms of process models or descriptions with detailed instructions for each type of activity of the integration process. 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 could 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, and describes how a model can be developed and validated (starting from scratch or using pre-defined partial models).

A.3.2.2 Human factor

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

The involving of company users is an important success factor for an integration project. It is considered that techniques used to build new manufacturing and service systems are currently difficult to understand for business users of the future system particularly in the domain of the Information Technology. Besides, according to the amount of investment necessary to build a new manufacturing and service system, one needs to be sure that the design solution of the new system meets the objectives defined in the initial user requirements. The design of the new system must be validated by users before any development or implementation.

The involving of people of the company will facilitate the final acceptance of the designed system and thus shorten the transition phase between the old and new systems. The methodology should make clear distinction between the two major phases of the design: user oriented design and technology oriented design. The experiences show that business people must be associated as much as possible in the user oriented design phase and as little as possible in detailed technology oriented design because it is an expert task (unless the technology considerations have a direct business effect).

The other aspect of human involvement in the enterprise is the place of humans in the designed entity, such as a plant.

To show the true place of the human in the implementation of the enterprise functions, there is a need to assign the appropriate ones of these tasks and functions developed in the Requirements Life-cycle Phase to the human element of the system. This can be done by considering the functional tasks as grouped in three boxes in the Preliminary Design Phase (see Figure A.11).

This action will separate the tasks of Mission Fulfilment and Management and Control as defined in the Requirements Analysis phase into three, thus assign the tasks or functions involved to the appropriate boxes which in turn define the automated information tasks which become the Information Systems Architecture functions and the automated manufacturing tasks which become the Manufacturing Equipment Architecture functions. The remainder (non-automated) become the functions carried out by humans as the Human and Organisational Architecture.

The split of functions for implementation between humans and machines forms the first definition of the implementation of the resulting manufacturing system. Because of the inclusion of humans, there must be three separate elements in the implementation scheme: the Information System Architecture, the Human and Organisational Architecture, and the Manufacturing Equipment Architecture.

Two lines, the Automatability Line and the Humanizability Line, can be defined giving the limits of automation and the limits of human involvement.

The Automatability Line shows the absolute extent of pure technologies in their capability to actually automate the tasks and functions. It is limited by the fact that many tasks and functions require human innovation, and so forth, and cannot be automated with presently available technology.

The Humanizability Line shows the maximum extent to which humans can be used to actually implement the tasks and functions. It is limited by human abilities in speed of response, breadth of comprehension, range of vision, physical strength, and so forth.

Still a third line, the Extent-of-Automation Line, can be drawn which shows the actual degree of automation carried out or planned in the subject enterprise entity. Therefore, it is the one which actually defines the boundary between the Human and Organisational Architecture and the Information Systems Architecture on the one side, and the boundary between the Human and Organisation Architecture and the Manufacturing (mission fulfilment) Equipment Architecture on the other side. Provided requirements such as timing and co-ordination are fulfilled, it makes no difference what functions are carried out by personnel versus machines, or what organisational structure or human-relations requirements are used. Therefore, the actual extent of automation is determined by political and human relations-based considerations as well as by technical ones. The location of the Extent of Automation Line is influenced by economic, political, social (customs, laws and directives, union rules), as well as technological factors.

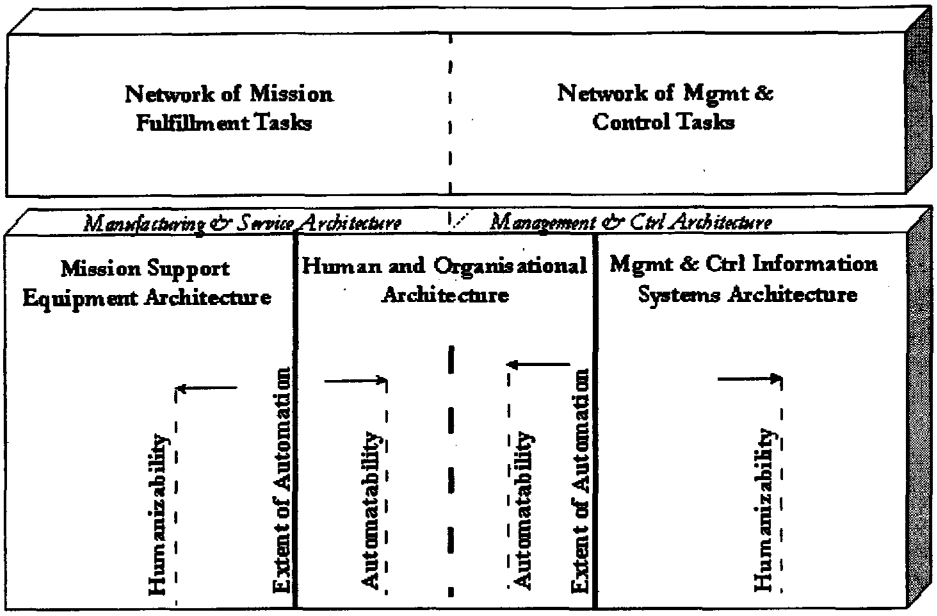


Figure A.11 – Introduction of the concepts of Automatability, Humanizability and Extent-of-Automation lines to define the three implementation architectures

A.3.2.3 Project management

In order to perform in an efficient way the analysis, the design and the implementation within an engineering and/or integration project, the methodology must associate with the available project management techniques in terms of project planning, project budgeting and control, project follow-up and so forth.

A logical separation can be made between a project life cycle and enterprise system life cycle (see A.3.1.3.4). Within the project life cycle

a) the control of the project is covered by the 'management and control' part of the project life cycle, and

b) the execution (operation) of the project is covered by the 'service to the customer' part,

as guided by the various phases defined in the life cycle of the system that is designed/built by the project. In this sense one of the main activities within the project 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.

Looking at the life history of a project it contains at least these three phases in time:

- project start-up, aimed at defining the project organisation (various teams and managers), project preparation (definition of the what, who, when and how), project planning and the organisation of the project start-up meeting;

- project control, aimed at acceptance of deliverables (hard and/or software, machines, various installations, etc.), monitoring of progress and continuous planning, managing problems and change, and executing reviews and auditing;

- project termination, aimed at the general acceptance and the final evaluation of the project.

Examples of project management approaches associated with a methodology can be found in GIM, SADT etc.

A.3.2.4 Economic aspects

A methodology must take the economic aspect into consideration. In fact, the choice of various investments depends on objectives that are often contradictory. To help designer to choose the best solution, both technical and economic views should be studied at the different steps of an integration project.

The methodology should allow the decomposition of the strategic objectives of the company into sub-objectives of each function; and the specification of the technical solution must be followed by a technical-economic evaluation. The economic evaluation can be split up in these 3 steps:

- calculation of the cost of the solution;
- performance measures of the solution;
- comparison of 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 will allow to economically validate or not the proposed solution.
Examples of technical-economic evaluation approach can be found in ECOGRAI, GEM (GRAI Evolution Methodology) or Activity Based Costing.

A.3.3 EMLs – Enterprise modelling languages

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

To develop enterprise models potentially more than one modelling language is needed. The situation is similar to software engineering where there are no known languages that span the needs of all models in all phases of the life cycle. The set of languages must be competent to express the models of all areas defined in the modelling framework of the Generalised Enterprise Reference Architecture, GERA.

Enterprise models must represent the enterprise operations from various modelling viewpoints (see A.3.1.5.3). For each area of the GERA modelling framework, there may be a modelling language selected according to the enterprise engineering methodology, which 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 then one area.

These two requirements must be satisfied in the definition of a complete set of enterprise modelling languages:

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

Any subject area of modelling may be covered with more than one language, the languages being of different 'expressive power', meaning that some languages may only be useful for the description of the subject area but not suitable for certain analysis tasks. For example, the languages that belong to the function view may differ in their capability of expressing certain characteristics of functions. For example, the dynamics of the function, the behaviour of the function, the subdivision of the function into function types such as product management, resource management, and co-ordination and planning. The necessary expressive power, and thus the selection of languages, is related to the methodology followed.

An enterprise engineering methodology may prescribe some analysis tasks that require a given modelling language. However, there should not be any prejudice built into the modelling languages as to what the methodology will be. It is necessary for any enterprise engineering methodology to have access to a *consistent set* of modelling languages for typical enterprise engineering tasks. Therefore, such consistent and complete sets must be selected or developed, e.g. CIMOSA set of languages, the choice of the set of languages by the GRAI methodology, etc.

Enterprise modelling languages may be defined as modelling constructs. Modelling constructs represent the different elements of the modelled enterprise entity and improve both modelling efficiency and model understanding. The form (representation) of modelling constructs has to be adapted to the needs of people creating and using enterprise models. Therefore, separate languages may exist to accommodate the aspects of different users (e.g. business users, system designers, IT-modelling specialists, and others). In addition, modelling languages may allow the formation of higher level constructs out of more basic constructs (macro constructs) to enhance modelling productivity.

Model based decision support and model driven operation control and monitoring require modelling constructs which are supporting the end users. They have to represent the operational processes according to the users' perceptions.

The semantics of the modelling languages may be described in terms of ontological theories (see A.3.4). This is especially important if enterprise models are to support the enterprise operation itself, because the models in that case must be executable. However, the definition of the formal semantics should be supported by natural language explanations of the concepts as well.

Examples of modelling languages can be found in ARIS (see reference [18] of A.4.1), CIMOSA (see references [8] and [14] of A.4.1), GRAI/GIM (see reference [15] of A.4.1), IEM (see reference [17] of A.4.1) or the IDEF family of languages (see references [8] and [16] of A.4.1). Relevant standards are: ENV 12 204, which defines a reference set of twelve Constructs for Enterprise Modelling (Business Process, Capability Set, Enterprise Activity, Enterprise Object, Event, Object View, Object State, Order, Organisational Unit, Product, Resource, Relation), and ISO 14258, which defines rules and concepts for enterprise models.

A.3.4 GEMCs – Generic enterprise modelling concepts

A.3.4.1 General

Generic enterprise modelling concepts are the most generically used concepts and definitions of enterprise engineering and modelling. The three forms of concept definition are, in increasing order of formality:

- glossaries;
- meta-models;
- ontological theories.

Some requirements that must be met are as follows:

- concepts defined in more than one form of the above must be defined in a mutually consistent way;
- those concepts which are used in an enterprise modelling language must also have at least a definition in the meta-model form, but preferably the definition should appear in an ontological theory.

A.3.4.2 Glossary

The terminology used in enterprise engineering can be defined in natural language as a glossary of terms. Such a glossary is a mandatory requirement for a complete generalised enterprise integration architecture and methodology. As a minimal requirement the glossary must define all concepts which are defined in the semi-formal meta-models and/or formal ontologies.

A.3.4.3 Meta-models

In the GERAM context, meta-models are conceptual models of the terminology component of modelling languages[10]. They describe the concepts used, their properties and relationships, as well as some basic constraints, such as cardinality constraints.

Meta-models are situated between informal and formal expressions. Normally, they are represented as an entity relationship schema or in a language similar in expressive power. The terminology defined in the integrated meta-schema may also be considered as the schema (at any one time) of an enterprise engineering tool's database of enterprise models.

A.3.4.4 Ontological theories

Ontological theories are formal models of the concepts that are used in enterprise representations. They capture rules and constraints of the domain of interest, allowing useful inferences to be drawn, to analyse, execute (e.g. for the purposes of simulation), cross check and validate models.

Ontological theories are a kind of generic enterprise model, describing the most generic aspects of enterprise-related concepts (function, structure, dynamics, cost, and so forth) and defining the semantics of the modelling languages used. They play a similar role to what 'data models' play in database design. Enterprise modelling languages backed by ontological theories (and their supporting enterprise engineering tools) provide the user with enhanced analysis capabilities.

Since separate enterprise modelling languages may be used to describe various aspects/views of the enterprise it is important to stress that the ontological models must be integrated, i.e. the language definitions for views should be views of an integrated meta-schema (if such a meta schema is defined) and/or of its underlying ontological model (if the corresponding ontological theory is defined). This purely technical requirement allows enterprise engineering tools to be used to cross-check the mutual consistency of enterprise models produced in the enterprise engineering process.[11]

A.3.5 PEMs – Partial enterprise models

A.3.5.1 General

Partial enterprise models (reusable reference models) are models which capture concepts common to many enterprises. PEMs will be used in enterprise modelling to increase modelling process efficiency. In the enterprise engineering process these partial models can be used as tested components for building particular enterprise models (EMs). However, in general such models still need to be adapted (completed) to the particular enterprise entity.

Partial models may be expressed as:

- models which capture some common part of a class of enterprises;

---

[10] Meta-models are models about models.

[11] There are theoretical limitations to this cross-checking, so the wording really should be to cross check to the best possible extent.

- paradigmatic (reference or prototypical) models which describe a typical enterprise of a class (prototype models can be subsequently modified to fit a particular case);
- abstract models of a part or whole of a class of enterprises which capture the commonalities but leave out specific details (this type of model is of the 'fill-in-the-blank' type).

A.3.5.2 Partial human role models

Needed are partial models on human roles in decision making, on professional capabilities and skills, on socio-technical aspects (motivation, incentives, and so forth). Related partial models will be on enterprise organisation and the identification of human responsibilities and authorisation in those models.

A.3.5.3 Partial process models

The provision of reusable reference models of business processes can significantly improve the efficiency of enterprise modelling. These models represent a common view of the enterprise's operational processes and are concerned with various processes, such as purchasing processes, order processes, product development processes, administrative processes (representing workflow procedures or CSCW), relations with external organisations (e.g. banks).

Partial process models could be tailored to specific industries or industry types (like automotive, electronic components industry, or more specific industries, such as car suspension manufacturing, VLSI manufacturing, and so forth). Or the models may represent typical management and control systems, such as various forms of enterprise co-operation. For example, the modern paradigms of extended and virtual enterprises could be represented as partial models guiding interested parties in defining their specific forms of co-operation.

It is to be noted, that these models of business processes would typically use one or more forms of model view (see A.3.1.5.3), such as function and behaviour models, database schemata, and so forth. Typically partial process models would describe common functionality but leave the definition of the process behaviour to the decision of the particular enterprise.

Partial models may be presented on various levels of abstraction and using various model content views; e.g. ISO 9000 quality models are partial models, defining typical or required policies that must be adopted by quality accredited companies (some ISO 9000 standards go further and define in more detail certain aspects of the business process functions). Many companies create partial models in form of company-wide database schemata, which are then enforced in all company databases, or are used as a basis for such designs. (Such common database schemata can be used as standard interfaces between processes.) Some partial models are provided as a system of model-fragments, which ensure that the resulting models define a high quality business process model as well as feasible system implementation.

A.3.5.4 Partial technology models

A.3.5.4.1 General

Partial technology models will provide common descriptions of resources and their aggregations like shop floors, assembly lines, flexible manufacturing systems, office systems, IT systems, etc. All of these partial models will most probably be industry and/or country specific, providing common descriptions of components (linked to supplier catalogues), common operational rules, etc.

3.5.4.2 Partial Models of IT systems

Partial models of IT systems can be all the components needed in communications and information processing, which will guide and enhance the design of information systems. This includes the enabling technology for enterprise engineering (EDI, STEP, HTML/WWW, etc.).

One important partial model commonly needed for enterprise engineering is the one of integrating services (see A.3.1.4.3) that provide portability across heterogeneous environments. These services have to include communication, processing/execution control, and presentation and information services. The definition of such services should itself refer to enabling standard definitions, such as: EDI; STEP; HTML/HTTP and all other communication protocols; CORBA-IDL; SQL3; Java services for execution, compilation, presentation etc.).

Such services can then be packaged in various ways as modular products or building blocks (see A.3.7).

A.3.6 EETs – Enterprise engineering tools

Enterprise engineering tools will deploy enterprise-modelling languages in support of enterprise engineering methodologies, and specifically must support the creation, use, and management of enterprise models.

Enterprise engineering tools must support the analysis and evaluation of the models (or descriptions) of the enterprise, and its products, as well as allow the enactment of the models for simulation. These functions are needed for design decision making in the course of enterprise engineering.

Engineering tools should provide user guidance for the modelling process and provide useful analysis capabilities for the use of the models in the enterprise engineering process. That is, the tools help the user utilise the models for the advancement of the engineering process to the best possible extent, as well as releasing the models for operation to support decision making and model based operation monitoring and control.

An enterprise-engineering tool is required to support the simultaneous design/engineering activity of the enterprise entity in question. Therefore it needs to

- support collaborative as well as individual design/engineering activity, and

- provide a shared design repository, or database, which allows the management of all partial and particular models and descriptions that are used or produced in the enterprise engineering process, including formal models and any other informal design descriptions, document, etc.

Depending on the enterprise entity in question these engineering tools of the enterprise may display a great variety. If the object of design is a project (i.e. project enterprise) or an enterprise (such as a company) then the tools will be supporting the creation of the design of such enterprise, including its business processes, resources, organisation, etc. If the enterprise entity in question is a product, or product type, then the tools will be supporting the design of the product, such as functionality, geometry, control system, operator procedures, and so forth.

Through the potential integration of the enterprise engineering and model execution services there is scope for the engineering services to be interconnected with enterprise management services. (For example, the initial simulation of a project's execution may use similar tools to those which are utilised for continuous planning of the project during project execution.)

Engineering tools should enable the user to connect the models with the real business process, so as to keep the models up-to-date. Note that engineering tools may be either separate or integrated with the model execution environment (see A.3.7).

The ideal engineering environment should be modular so that alternative methodologies could be based on it. Therefore, engineering tools should provide an environment which is extensible rather than be based on a closed set of models, leaving space for alternative modelling methods (e.g. through enriching the modelling language constructs, or adding new views, as appropriate).

Some examples of engineering tools based on modelling languages (when the enterprise entity in question is the enterprise or a project enterprise itself) are: ARIS Toolset (ARIS, see reference [18] of A.4.1), FirstSTEP (CIMOSA), MOGO (IEM, see reference [19] of A.4.1), KBSI Tools (see reference [20] of A.4.1), METIS, etc. There are many examples for engineering tools of the enterprise for the case when the enterprise entity in question is a product; such tools include tools for product modelling and design, simulation, visualisation, control systems design, and so forth; e.g. STEP Tools.

A.3.7 EMOs – Enterprise modules

Enterprise modules are implemented building blocks or systems (products, or families of products) that can be utilised as common resources in enterprise engineering and enterprise integration. As physical entities (systems, subsystems, software, hardware, and available human resources/professions) such modules are accessible in the enterprise, or can be made easily available from the market place. In general EMOs are implementations of partial models identified in the field as the basis of commonly required products for which there is a market. Enterprise modules may be offered as a set, such that if the design of the enterprise is following the partial models that form the basis of this set, then the resulting particular enterprise's business system can be implemented using some or all modules of this set of modules. One set of enterprise modules of distinguished importance is the Integrating Infrastructure that implements the required Integrating IT Services (see A.3.5.4.2).

A.3.8 EMs – Enterprise models

The goal of enterprise modelling is to create and continuously maintain a model of a particular enterprise entity. A model should represent the reality of the enterprise operation according to the requirements of the user and his application. This means the granularity of the model has to be adapted to the particular needs, but still allows interoperability with models of other enterprises. Enterprise models include all those descriptions, designs, and formal models of the enterprise which are prepared in the course of the enterprise's life history.

Enterprise models are expressed in enterprise-modelling languages and are maintained (created, analysed, stored, distributed) using enterprise engineering tools. Both model creation and model use should be supported by real-time information services. The use of such services will ensure access to real time information in both enterprise environments, the engineering and the operational one.

Some important uses of enterprise models are:

- decision support for evaluating operational alternatives in the enterprise engineering process (enabling operation analysis and capturing the results of synthesis);
- communication tool which enables the mutual understanding of issues between stakeholders of the enterprise, both internal and external ones;
- model driven operation control and monitoring, for efficient business process execution, and training of new personnel, where enterprise models serve as demonstration of the real business process for new employees.

A.3.9 EOSs – Enterprise operational systems

Enterprise Operational Systems support the operation of a particular enterprise. They consist of all the hardware and software needed to fulfil the enterprise objective and goals. Their contents are derived from enterprise requirements and their implementation is guided by the design models which provide the system specifications and identify the enterprise modules used in the system implementation.

A.4 Glossary of references

A.4.1 General References

[1] Charles J. Petrie, Jr., Enterprise Integration Modelling; ICEIMT Conference Proceedings, The MIT Press, Cambridge, MA, (1992) (ISBN 0-262-66080-6).

[2] CIMOSA – Open System Architecture for CIM, ESPRIT Consortium AMICE, Springer-Verlag, Berlin, 1993, (ISBN 3-540-56256-7), (ISBN 0-387-56256-7).

[3] T.J. Williams, et al., Architectures for integrating manufacturing activities and enterprises, Computers in Industry – Vol. 24, Nrs 2-3 (1994).

[4] D. Shorter, Editor, An evaluation of CIM modelling constructs – Evaluation report of constructs for views according to ENV 40 003, Computers in Industry – Vol. 24, Nrs 2-3 (1994).

[5] P. Bernus, and L. Nemes, A Framework to Define a Generic Enterprise Reference Architecture and Methodology, Proceedings of the International Conference on Automation, Robotics and Computer Vision (ICARCV'94), Singapore, November 10–12, (1994). also in *Computer Integrated Manufacturing Systems* 9,3 (July 1996) pp 179-191.

[6] T.J. Williams and Hong Li, A Specification and Statement of Requirements for GERAM (The Generalised Enterprise Reference Architecture and Methodology) with all Requirements illustrated by Examples from the Purdue Enterprise Reference Architecture and Methodology PERA, Research Report 159, Purdue Laboratory for Applied Industrial Control, (November 1995), Version 1.1.

[7] CIMOSA – Open System Architecture for CIM, Technical Baseline; Version 3.2 CIMOSA Association, private publication (March 1996).

[8] F.B. Vernadat, "Enterprise Modelling and Integration: principles and applications", Chapman & Hall, London, (1996), (ISBN 0-412-60550-3).

[9] K. Kosanke, "Comparison of Enterprise Modelling Methodologies", Proceedings DIISM'96, Chapman & Hall, London, (1997).

[10] M.S. Fox, "The TOVE project: towards a common-sense model of the enterprise", Proc ICIEMT92, Petrie, C.J., Jr. (Editor), MIT Press, (1992) pp 310-19.

[11] K. Kosanke, J. Nell (Editors), Proceedings of the International Conference on Enterprise Integration and Modelling Technology, Springer-Verlag, (1997).

[12] R.H. Weston, "Workbenches and reference models for enterprise engineering" chapter in *Handbook of Life Cycle Engineering: Concepts, Tools and Techniques*, Edited by A. Molina, J.M. Sanchez, A. Kusiak, to be published by Chapman & Hall, London, (1998).

[13] P. Bernus, K. Mertins and G. Schmidt, Handbook on Architectures of Information Systems, Springer-Verlag, (1998).

[14] F.B. Vernadat, The CIMOSA Languages, in P. Bernus, K. Mertins and G. Schmidt, Handbook on Architectures of Information Systems, Springer-Verlag, (1998), pp243-264.

[15] G. Doumeingts, B. Vallespir, D. Chen, Decisional Modelling using the GRAI Grid, in P. Bernus, K. Mertins and G. Schmidt, Handbook on Architectures of Information Systems, Springer-Verlag, (1998), pp313-338.

[16] C. Menzel, R.J. Mayer, The IDEF Family of Languages, in P. Bernus, K. Mertins and G. Schmidt, Handbook on Architectures of Information Systems, Springer-Verlag, (1998), pp209-242.

[17] G. Spur, K. Mertins, R. Jochem, Integrierte Unternehmensmodellierung, Beuth Verlag, Berlin (1993).

[18] A.W.Scheer, ARIS, in P. Bernus, K. Mertins and G. Schmidt, Handbook on Architectures of Information Systems, Springer-Verlag, (1998), pp541-566.

[19] K. Mertins, R.Jochem, MO^2GO, in P. Bernus, K. Mertins and G. Schmidt, Handbook on Architectures of Information Systems, Springer-Verlag, (1998), pp589-600.

[20] F.Tissot, W.Crump, A Truly Integrated Enterprise Modeling Environment, in P. Bernus, K. Mertins and G. Schmidt, Handbook on Architectures of Information Systems, Springer-Verlag, (1998).

A.4.2 Standards

ENV 40 003, *Computer Integrated Manufacturing — Systems Architecture — Framework for Enterprise Modelling,* CEN/CENELEC, 1990.

ENV 12 204, *Advanced Manufacturing Technology — Systems Architecture — Constructs for Enterprise Modelling,* CEN/TC 310/WG1, 1995.

CEN/TC 310, *CIM Systems Architecture — Enterprise model execution and integration services — Evaluation report,* CEN Report CR 1831, 1995.

CEN/TC 310, *CIM Systems Architecture — Enterprise model execution and integration services — Statement of requirements,* CEN Report CR 1832, 1995.

ISO 14258: 1998, *Industrial automation systems - Concepts and rules for enterprise models* (ISO/TC 184/SC 5/WG 1).

ISO 14258, *Industrial automation systems — Concepts and rules for enterprise models: Technical Corrigendum 1.*

Annex B
(informative)

Bibliography

B.1 CIMOSA references

AMICE, ESPRIT Project 688, "Open System Architecture for CIM," Springer-Verlag, Berlin, 1988.

AMICE Consortium, "Open System Architecture for CIM," Research Report of ESPRIT Project 688, Vol. 1, Springer-Verlag, Berlin, 1989.

AMICE Consortium, "Open System Architecture, CIMOSA, AD 1.0, Architecture Description," ESPRIT Consortium AMICE, Brussels, Belgium, 1991.

AMICE Consortium, "CIMOSA, Architecture Description," ESPRIT Project 5288, Milestone M-2, AD2.0, 2, Document RO443/1, Brussels, Belgium, August 24, 1992.

Beeckman, Dirk, "CIMOSA: Computer Integrated Manufacturing — Open System Architecture," *International Journal of Computer Integrated Manufacturing,* Vol. 2, pp. 94–105, 1989.

CIMOSA Association, "CIMOSA – Open System Architecture for CIM," Technical Baseline, Version 3.2, private publication, March 1996.

Gransier, T., and Schonewolf, W. (Editors), Special Issue: "Validation of CIMOSA," *Computers in Industry,* Vol. 27, No. 2, pp. 95-213, October 1995. Devoted exclusively to CIMOSA-Based Discussions.

Jorysz, H.R. and Vernadat, F.B., "CIMOSA Part 1: Total Enterprise Modelling and Function View," *International Journal of Computer Integrated Manufacturing,* Vol. 3, pp. 144–156, 1990.

Jorysz, H.R. and Vernadat, F.B., "CIMOSA Part 2: Information View," *International Journal of Computer Integrated Manufacturing,* Vol. 3, pp. 157–167, 1990.

Klittich, M., "CIMOSA Part 3: CIMOSA Integrated Infrastructure — the Operational Basis for Integrated Manufacturing Systems," *International Journal of Computer Integrated Manufacturing,* Vol. 3, pp. 168–180, 1990.

Vernadat, F., "Modelling CIM Enterprise with CIMOSA," in *Proceedings of Rensselaer's Second International Conference on Computer Integrated Manufacturing,* pp. 236-243, May 1990.

Vlietstra, Jacob, "An Open System Architecture in Computer-Integrated Manufacturing: CIMOSA," *Journal of Applied Manufacturing Systems,* pp. 23–35, Summer 1991.

B.2 GRAI-GIM references

Chen, David, "GRAI Integrated Framework for Enterprise Integration," Minutes, Eleventh Workshop Meeting, IFAC/IFIP Task Force on Architectures for Enterprise Integration, Galway, Ireland, April 18-19, 1995.

Doumeingts, G., *Methode GRAI: Methode de Conception des Systems en Productique,* Thesis d'etat: Automatique: Universite de Bordeaux I, France, November 13, 1984.

Doumeingts, G., Vallespir, B., Darracar, D., and Roboam, M., "Design Methodology for Advanced Manufacturing Systems," *Computers in Industry,* Vol. 9, pp. 271–296, 1987.

Doumeingts, G., Vallespir, B., and Marcotte, F., "A Proposal for an Integrated Model of a Manufacturing System: Application to the Re-engineering of an Assembly Shop," *Control Engineering* Practice, Vol. 3, No. 1, pp. 67, 1995.

Doumeingts, G., Vallespir, B., Zanettin, M., and Chen, D., *GRAI-GIM Integrated Methodology, A Methodology for Designing CIM Systems*, Version 1.0, LAP/GRAI, University Bordeaux I, France, May 1992.

B.3 PERA references

Li, Hong, and Williams, T.J., *A Formalization and Extension of the Purdue Enterprise Reference Architecture and the Purdue Methodology,* Technical Report 158, Purdue Laboratory for Applied Industrial Control, Purdue University, W. Lafayette, IN, December 1994.

Rathwell, G.A., and Williams, T.J., "Use of the Purdue Enterprise Reference Architecture and Methodology in Industry (the Fluor Daniel Example)," in *Modeling and Methodologies for Enterprise Integration* (P. Bernus and L. Nemes, Editors), Chapman and Hall, London, 1996.

Williams, T.J., "The Purdue Enterprise Reference Architecture," *Computers in Industry,* Vol. 24, No. 2-3, pp. 141-158., 1994.

Williams, T.J., Rathwell, G.A., and Li, Hong (Editors), *A Handbook on Master Planning and Implementation for Enterprise Integration Programs,* Report 160, Purdue Laboratory for Applied Industrial Control, Purdue University, W. Lafayette, IN, 1996.

Williams, T.J., *The Purdue Enterprise Reference Architecture,* Instrument Society of America, Research Triangle Park, NC, 1992. Purdue Reports are available from Purdue University (http://www.ecn.purdue.edu/IIES/PLAIC).

B.4 GERAM references

Bernus, Peter and Nemes, Laszlo, "A Framework to Define a Generic Enterprise Reference Architecture and Methodology," minutes, Eighth Workshop Meeting, IFAC/IFIP Task Force on Architectures for Enterprise Integration, Vienna, Austria, June 11-12, 1994.

Bernus, Peter and Nemes, Laszlo, "A Framework to Define a Generic Enterprise Reference Architecture and Methodology," minutes, Ninth Workshop Meeting, IFAC/IFIP Task Force on Architectures for Enterprise Integration, Ottawa, Canada, August 25–26, 1994; also in *Proceedings of the International Conference on Automation, Robotics and Computer Vision (ICARCV'94),* Singapore, November 10–12, 1994.

Bernus, Peter and Nemes, Laszlo, "Enterprise Integration, Engineering Tools for Designing the Business Process of Enterprise," paper presented at the Australian-Taiwan Joint Workshop on Materials, Wollongong East, NSW, Australia, June 26–27, 1995; minutes, Twelfth Workshop Meeting, IFAC/IFIP Task Force on Architectures for Enterprise Integration, Purdue University, West Lafayette, Indiana, USA, June 19–21, 1995.

Bernus, Peter and Nemes, Laszlo, *A Framework to Define a Generic Enterprise Reference Architecture and Methodology,* Divisional Report Number: MTM 366, CSIRO Division of Manufacturing Technology, Preston, Victoria, Australia 3072, undated. Also in minutes, Tenth Joint Meeting, IFAC/IFIP Task Force on Architectures for Enterprise Integration, Singapore, November 7–8, 1994, Grenoble, France, December 13, 1994.

Williams, T.J., "Contributions of the Purdue Enterprise Reference Architecture and Methodology (PERA) to the Development of a General Enterprise Reference Architecture and Methodology (GERAM)," in *Proceedings of the Third International Conference on Automation, Robotics and Computer Vision (ICARCV'94)*, pp. 83–87, Singapore, November 9, 1994.

Williams, T.J. and Li, Hong, " Proposed Three-Dimensional Model of GERAM," minutes, Tenth Joint Workshop Meeting, IFAC/IFIP Task Force on Architectures for Enterprise Integration, Grenoble, France, December 13, 1995.

Williams, T.J., and Li, Hong, *A Specification and Statement of Requirements for GERAM,* Report 159, Purdue Laboratory for Applied Industrial Control, Purdue University, West Lafayette, IN, September 1995.

B.5 References on the work of the IFAC/IFIP Task Force

Bernus, P., and Nemes, L, *Modelling and Methodologies for Enterprise Integration,* Chapman and Hall, London, 1996. Contains papers on all the architectures discussed and related research by Task Force members and others.

Bernus, P., Nemes, L., and Williams, T.J. (Editors), *Architectures for Enterprise Integration,* Chapman and Hall, London, 1996. Revised book version of Williams (1993).

Williams, T.J. (Editor), *Architectures for Integrating Manufacturing Activities and Enterprises,* Technical Report, IFAC/IFIP Task Force, Purdue University, West Lafayette, IN, March 1993, revised May1993.
Williams, T.J., Bernus, P., Brosvic, J., Chen, D., Doumeingts, G., Nemes, L., Nevins, J.L., Vallespir, B., Vlietstra, J., and Zoetekouw, D., "Architectures for Integrating Manufacturing Activities and Enterprises," *Proceedings, Twelfth IFAC Congress,* Sydney, Australia, July 19-23, 1993; Chapman and Hall, London (1994); also presented at the IFIP WG5.3 Workshop on Toward World Class Manufacturing, Litchfield Park, Arizona, Sept. 13-16, 1993, and at the JSPE-IFIP WG5.3 Workshop on *The Design of Information Infrastructure Systems for Manufacturing,* Tokyo, Japan, Nov. 8-10, 1993. It was published in the *Proceedings* of both workshops as published by Elsevier-North Holland for IFIP.

Williams, T.J., Bernus, P., Brosvic, J., Chen, D., Doumeingts, G., Nemes, L., Nevins, J.L., Vallespir, B., Vlietstra, J., and Zoetekouw, D., "Architectures for Integrating Manufacturing Activities and Enterprises," *Computers in Industry,* Vol. 24, No.2-3, pp. 111-139, 1994; *Control Engineering Practice,* Vol. 2, No. 6, pp. 939-960, 1994.

B.6 Other important references in the field of enterprise integration

Kosanke, K., and Nell, J.G. (Editors), *Enterprise Engineering and Integration: Building International Consensus*, Springer Verlag, 1997 (ISBN 3-540-63402-9).

Petrie, C.J., Jr. (Ed.), "Enterprise Integration Modelling," *Proceedings of the First International Conference on Enterprise Integration Modelling Technology,* MIT Press, 1992.

Scheer, A-W, *Architecture for Integration Information System*, Berlin, 1992.

Scheer, A-W, *Enterprise-wide Data Modelling -- Information Systems in Industry,* Springer Verlag, Berlin-Heidelberg, 1989.

Spur, G., Mertins, K., Jochem, R., *Integrated Enterprise Modelling,* Beuth-Verlag, Berlin, 1996.

Vernadat, F.B., *Enterprise Modelling and Integration,* Chapman and Hall, London, 1996. A general text of the field but presenting much on CIMOSA.

ICS 25.040.01

Price based on 43 pages