GERAM:

Generalised Enterprise Reference Architecture and Methodology

Version 1.6.3

IFIP–IFAC Task Force on

Architectures for Enterprise Integration

March 1999

Permission granted to publish GERAM V1.6.3 as Annex to
ISO WD15704, Requirements for enterprise-reference architectures and methodologies

Document History:

1. Bernus/ Nemes document GERAM definition
2. Kosanke/Vernadat/document GERAM definition
3. Williams document GERAM definition
4. 1,2,3 merged by Kosanke/Vernadat
5. 1,2,3 merged by Bernus/Nemes
6. 4,5 merged by Kosanke sent to Vernadat
7. revisions added to 6 by Vernadat and sent to Bernus/Kosanke
8. revisions added to 7 by Bernus and sent to Vernadat/Kosanke/Nemes
9. revisions merged by Kosanke with accepted revision incorporated in 9 and to be distributed to Task Force members for comment on WWW (Bernus to do the conversion to html and place the document to the web for discussion in email on the Task Force mailing list)
10. feedback from Task Force members (T.J. Williams, D. Shorter) discussed by P. Bernus and K. Kosanke (input from T.J. Williams only) and document updated accordingly.
11. Revision reflects feedback received from WG1 and Task Force Meeting, Paris 97-06-25/27, added: input from D. Chen and R. Weston (consolidated by K. Kosanke).
12. Bernus extended section 3.6 on Enterprise Engineering Tools.
13. GERAM Version 1.5 Distributed to Task Force Members to get their approval to include GERAM V1.5 with the ISO TC184/SC5/WG1 GERAM Requirements Draft International Standard as an appendix.
14. Ted Williams produced version N405 from earlier version.
15. Colorado meeting decided to update V1.5 with changes required by the evolving text of the GERAM requirements standard and to identify in an email discussion before the May 1998 ISO TC184/SC5/WG1 meeting which N405 additions need to be incorporated into V1.5.
16. Email discussion by active task force members proposed some changes, including the lifting of sections from ISO TC184/SC5/WG1 N405 , which was a version edited by Ted Williams on the basis of a former version of this document, and inserting these into V1.5
17. On the basis of 15. and 16. Bernus and Kosanke edits and proposes Version 1.6.0
18. Members voted in April-May 1998 and accepted V1.6.0 as the TF definition of GERAM
19. On response by two members minor corrections were made by Bernus to V1.6.0 (These were: making the figures consistent in their representation of the subdivision of design into preliminary and detailed design phases, lifting Fig.11 of N405 back to the document to better illustrate the text, stressing the difference between mission fulfilment technology equipment and management and control support technology equipment in Fig.9, as well as tidying figures, and some trivial corrections. This version is called V1.6.1 Only Williams and Kosanke poined out som minor points in v1.6.1. Further final editorial changes were made by Bernus and V1.6.2 on demand from Kosanke and Williams.
20. GERAM Version 1.6.2. is submitted to ISO as the Task Force definition of GERAM. Since V1.6.1 and V1.6.2. did not add new technical detail to V1.6.0 it is therefore construed to have been accepted by the Task Force, on the basis of the vote on V1.6.0.
21. Based on David Shorter's comments Peter Bernus made editorial changes to V1.6.2 and produced Version 1.6.3. Also in this version figures have been changed to black and white. The changes have been placed on the WWW for Task Force members to inspect and raise objections if in their opinion there was any change of non-editorial nature. (The intention is that any issues on which there has not ben any agreement should be addressed in a later version).

Table Of Contents

1 Introduction

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 [7], by the GRAI Laboratory on GRAI and GIM [15], and by the Purdue Consortium on PERA [21], (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 integration 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 integration 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.

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.

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 integration, those based on product models and those based on business process design. It also offers new insights into the project management of enterprise integration 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

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.

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.

2 The Framework for Enterprise Engineering and Enterprise Integration

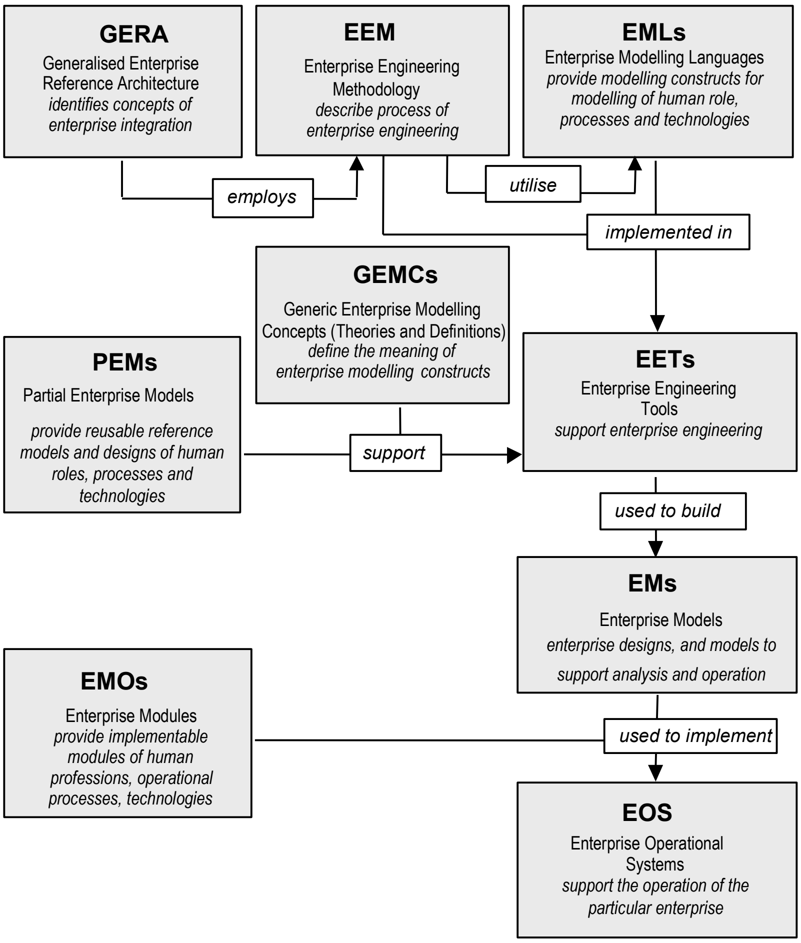

Figure 1 - GERAM (Generalised Enterprise Reference Architecture and Methodology) framework components

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 Fig. 1 and is briefly described in the following. Each component is then defined in more details in Section 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) that 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) that 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 that 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.

The operational use of enterprise models is supported by specific modules (EMOs) that 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 integration methodology and its proposed supporting components.

2.1 Definition of GERAM Framework Components

2.1.1 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).

2.1.2 EEMs - Enterprise Engineering Methodology

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.

2.1.3 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.

2.1.4 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.

2.1.5 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 model 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 I Reference Architectures'[3]. These terms have the same meaning.

2.1.6 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.

2.1.7 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.

2.1.8 EMOs - Enterprise Modules

EMOs are products that 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.

2.1.9 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.

3 Description of GERAM Framework Components

3.1 GERA – Generalised Enterprise Reference Architecture

3.1.1 General

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

a) Human oriented concepts: They cover 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

[3] Life-cycle architectures such as GERA are referred to as 'Reference Architectures of type II'

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 that 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 that promote 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: They deal with enterprise operations (functionality and behaviour) and cover 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: They deal 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 DIS 14258 defines Rules and Guidelines for Enterprise Models.

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.

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.

Often humans and groupings of humans will be assigned a number of roles and responsibilities that 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 harmo-

---

[4] ARIS: Architektur fur Informations Systeme (Architecture for Information Systems)

[5] CIMOSA: CIM Open Systems Architecture

[6] GRAI/GIM: Graphe Résultats et Activitées Interreliées (Graphs with Results and Activities Interrelated)/ GRAI Integrated Methodology

[7] IEM: Integrated Enterprise Modelling

[8] PERA: Purdue Enterprise Reference Architecture

nises 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 that 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 re-used 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 that 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;
- improve its resilience to the loss of core competancies in the form of knowledgeable human assets.

3.1.2.1 Human Role Models

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, ...), skill models, 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 3.1.4.1). 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 3.2).

3.1.3 Process oriented concepts

Business process-oriented modelling aims at describing the processes in the enterprise capturing both their functionality (that is what has to be done) 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 in the following sections.

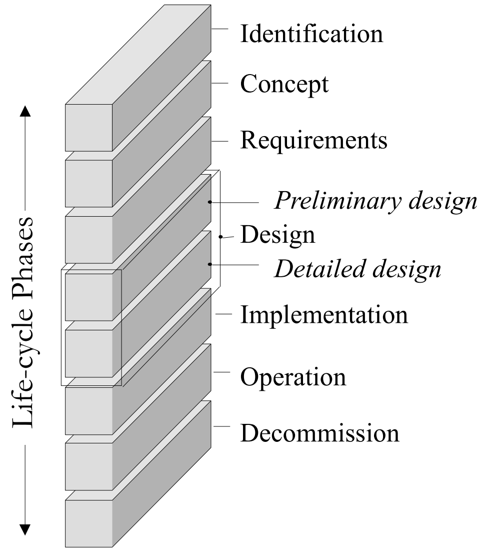

Figure 2 – GERA Life-cycle phases for any enterprise or entity

3.1.3.1 Life-cycle

Figure 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, that my be subdivided further as demonstrated for the design type activities that 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.

3.1.3.1.1 Entity Identification

This is the set of activities that identifies 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 that define what is the entity of which the life-cycle is being considered.

3.1.3.1.2 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.

3.1.3.1.3 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).

3.1.3.1.4 Entity Design

The activities that 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. Example in Fig. 2: dividing design into functional design or specification and detailed design to permit the separation of

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

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

3.1.3.1.5 Entity Implementation

The activities that define all those tasks that 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.

3.1.3.1.6 Entity Operation

The activities of the entity that 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.

3.1.3.1.7 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.

---

[9] Note that a) the need for such subdivision is found methodologically very important (see the Purdue Guide for Master Planing), 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.

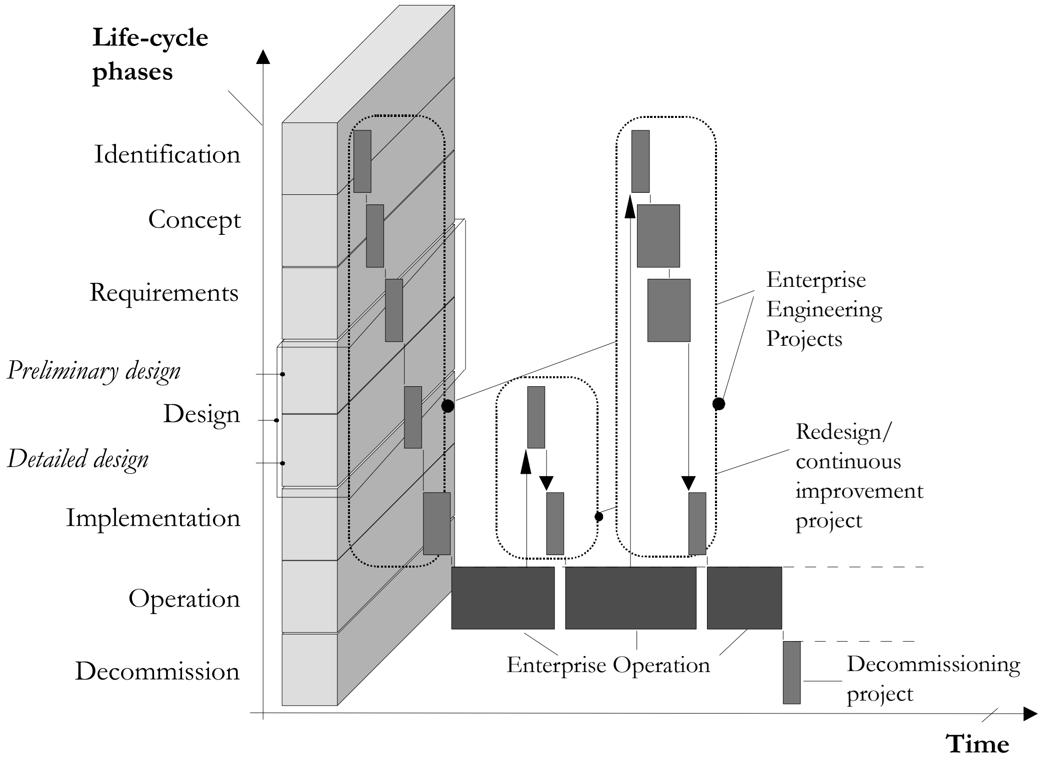

Figure 3 – Parallel processes in the entity's life-history

3.1.3.2 Life history

The life history of a business entity is the representation in time of tasks carried out on the particular entity during its entire life span. Relating to the life-cycle 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.

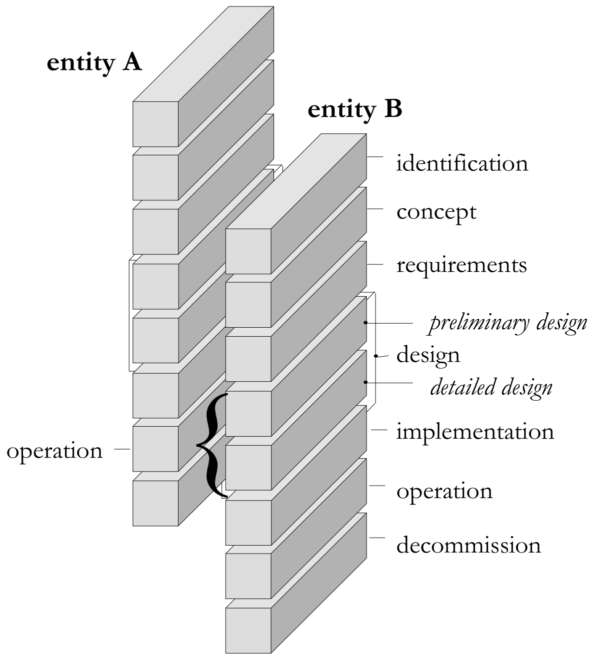

Figure 4 – Example for the relationship between life-cycles of two entities

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 that in turn rely on the same type of life-cycle activities as defined in the GERA life-cycles (3.1.2.1). For this reason life-cycle activities are a useful abstraction in understanding the life history of any entity.

Figure 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.

3.1.3.3 Entity types in Enterprise Integration

Figure 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.

3.1.3.3.1 Operation oriented Enterprise Entity Types

These enterprise entity types are all concerned with different types of operations

3.1.3.3.1.1 Project Enterprise Entity (Type A):

This type defines an enterprise (often with a short life history) that 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 that 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).

3.1.3.3.1.2 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 that are enterprises themselves, e.g. project enterprises are regularly created by engineering and building companies.

3.1.3.3.1.3 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.

3.1.3.3.2 Recursive Enterprise Entity Types

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

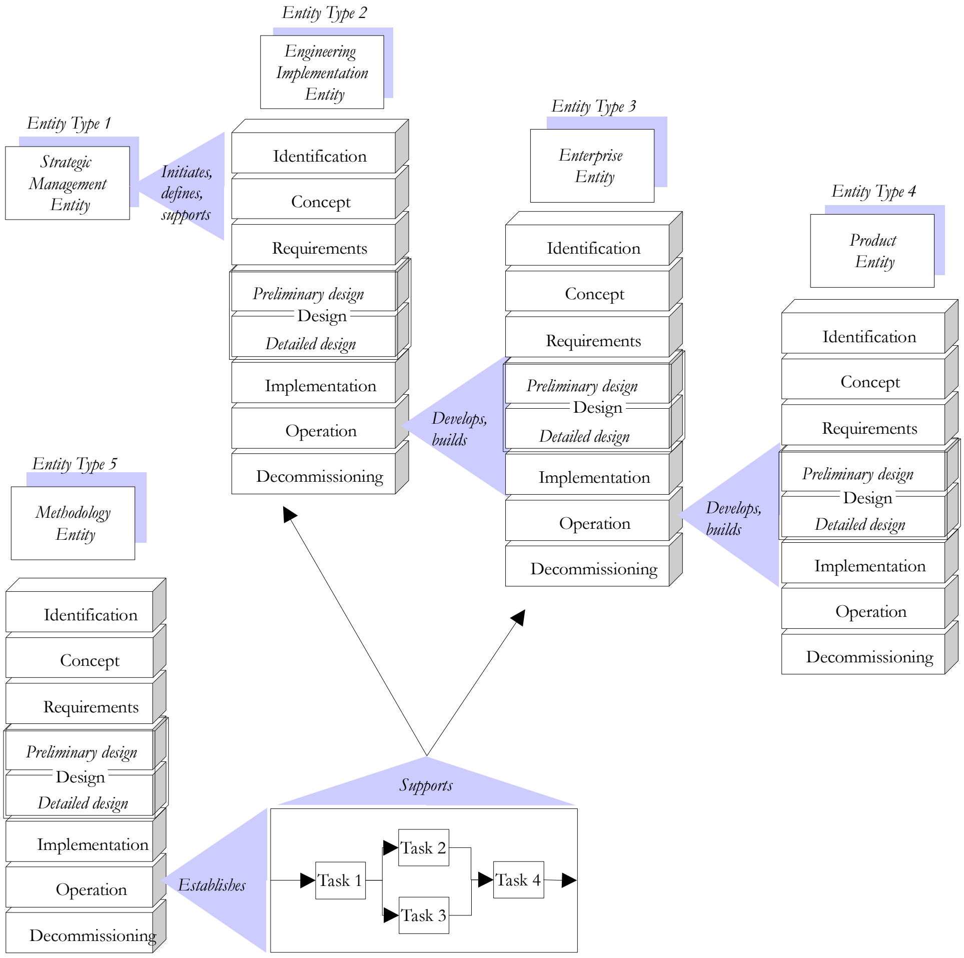

Figure 5 –Relationships between life-cycles of GERA entity Types

- *Strategic Enterprise Management Entity (Type 1):* defines the necessity and the starting of any enterprise engineering / integration effort.
- *Enterprise Engineering/Integration Entity (Entity Type 2):* 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):* 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):* 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):* 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 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 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 (that 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.

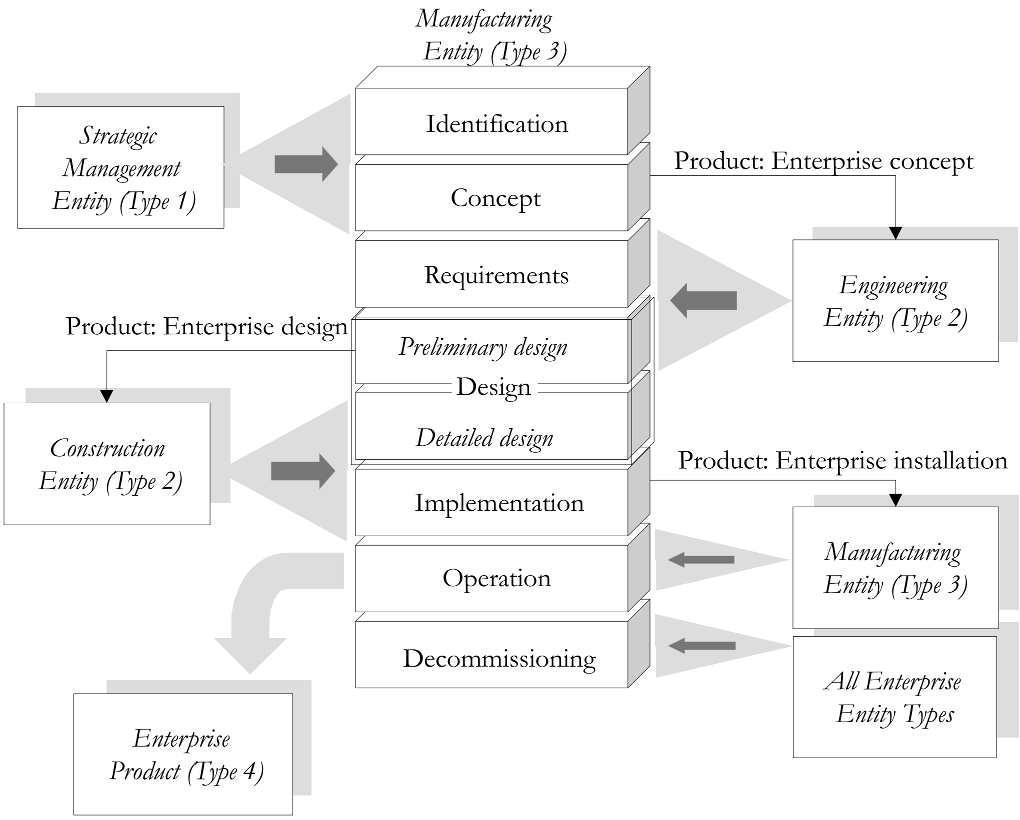

Figure 6 –Relationships between GERA entity Types

The operational relations of the different entity types are also shown in Figure 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 3 in which the engineering processes could relate to an Entity Type 2 and the operational processes to an Entity Type 3 that 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.

3.1.3.4 Process Modelling

Process modelling is the activity that results in various models of the management & control as well as the service and production *processes*, and their relationships to the resources, organisation, products etc. of the enterprise. Process modelling allows 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.

3.1.4 Technology oriented concepts

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).

3.1.4.1 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.

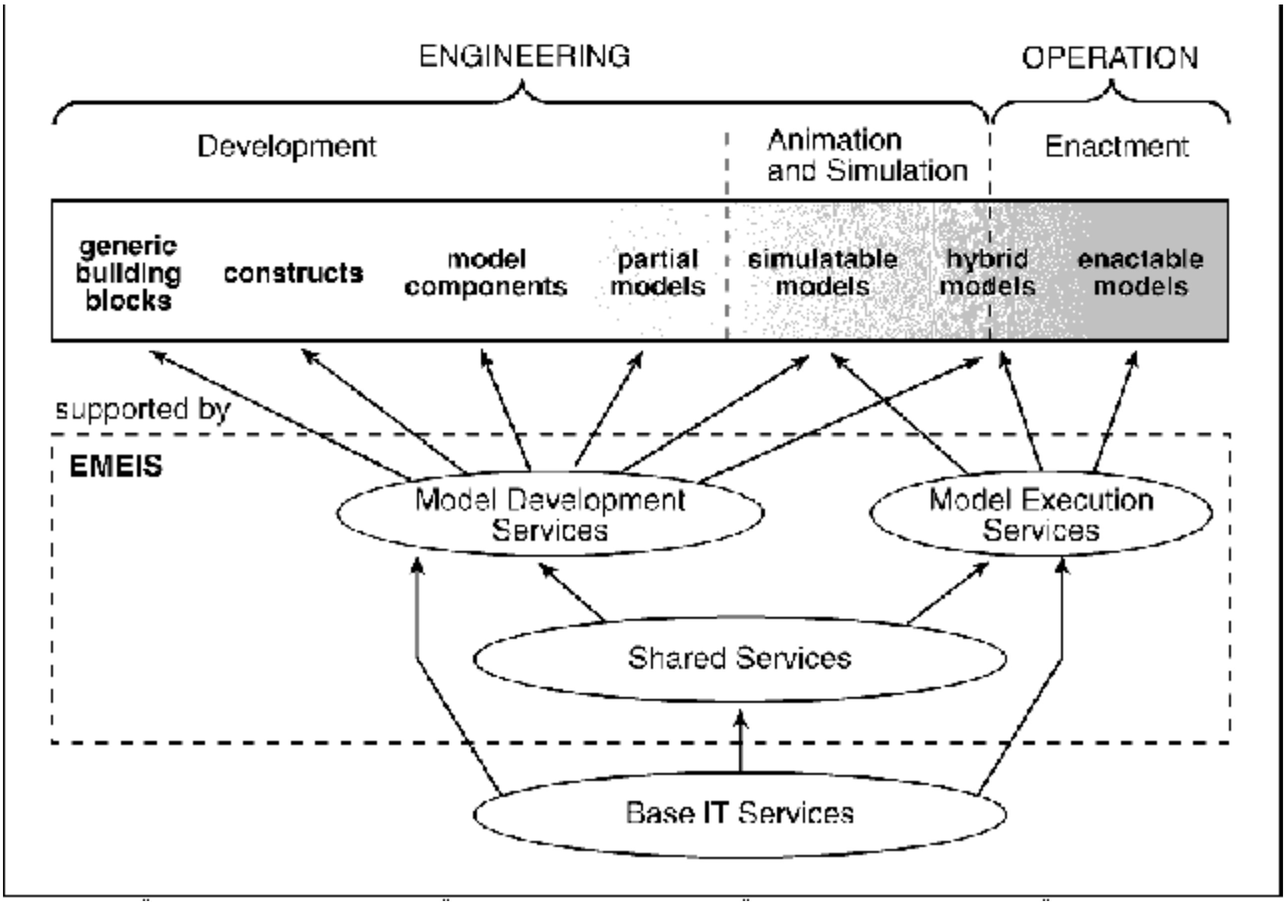

Figure 7 – Reference Model of EMEIS

3.1.4.2 Enterprise Model Execution and Integration Services (EMEIS)

To illustrate the potential use of computer executable models for on-line operation of the enterprise, Figure 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, CIMBIOSIS, CIMOSA, MIDA, OPAL, PISA, TOVE). Some of these project results have been evaluated in a CEN/TC310 report and have lead to statement of requirements for enterprise model execution and integration services by CEN/TC310 as well. The statement of requirements distinguishes between the model development services (MDS), the model execution services (MXS) and the general IT services (see Fig. 7). However no explicit service entities have been defined.

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

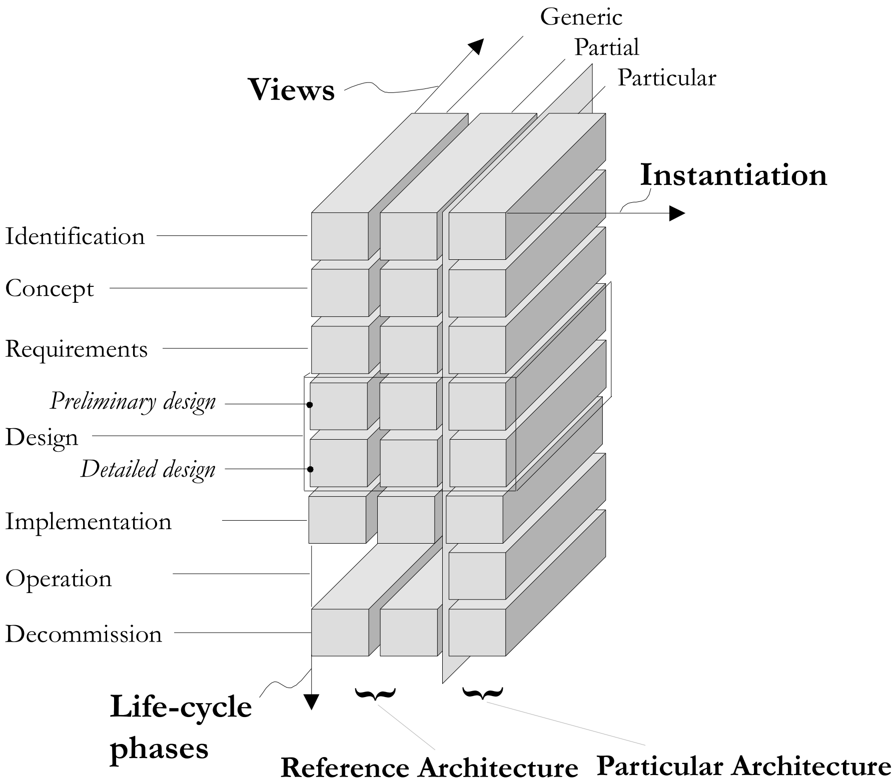

Figure 8 – The GERA Modelling Framework (The left hand side represents the reference models the right hand side the resulting particular enterprise models.)

3.1.5 Modelling Framework of GERA

GERA provides an analysis and modelling framework that is based on the life-cycle concept and identifies 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 8 shows the three dimensional structure identified above that 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.

3.1.5.1 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 [13].

The emphasis in enterprise modelling is currently on process and product models for representing enterprise operations. Process oriented modelling allows 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.

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 that represent part of the model that 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 previously practised. This modelling activity is, however, over and above the already practised ones, not to replace them

3.1.5.2 View Concepts

To decrease the apparent complexity of the resulting enterprise models GERA provides the view concept that 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 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 Contents Views: function, information, resource, organisation;
- Entity Purpose Views: customer service and product, management and control;
- Entity Implementation View: human implemented tasks, automated tasks (mission support technology, and management and control 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 are 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 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

a) to express a design choice;

b) to simulate a process in order to find out 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.

3.1.5.2.1 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, decisional 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, such as CIMOSA [8,14] function view models, GRAI Grid [15] and GRAI Net representations of decision centres, Petri nets, Event Driven Process Chains, Generalised Process Networks, QGERT or GPSS models, ... and so on.

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, 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 cover a great deal. This is because there are many different languages that fit any given model view in this category.

3.1.5.2.2 Entity Purpose Views

Two different views 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 that 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.

3.1.5.2.3 Entity Implementation View

The implementation of the enterprise entity may be presented in two different views based on the division between human- and automated tasks:

- The *Human Activities View* 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 humanisability) and those that will be done by humans (extent of automation).

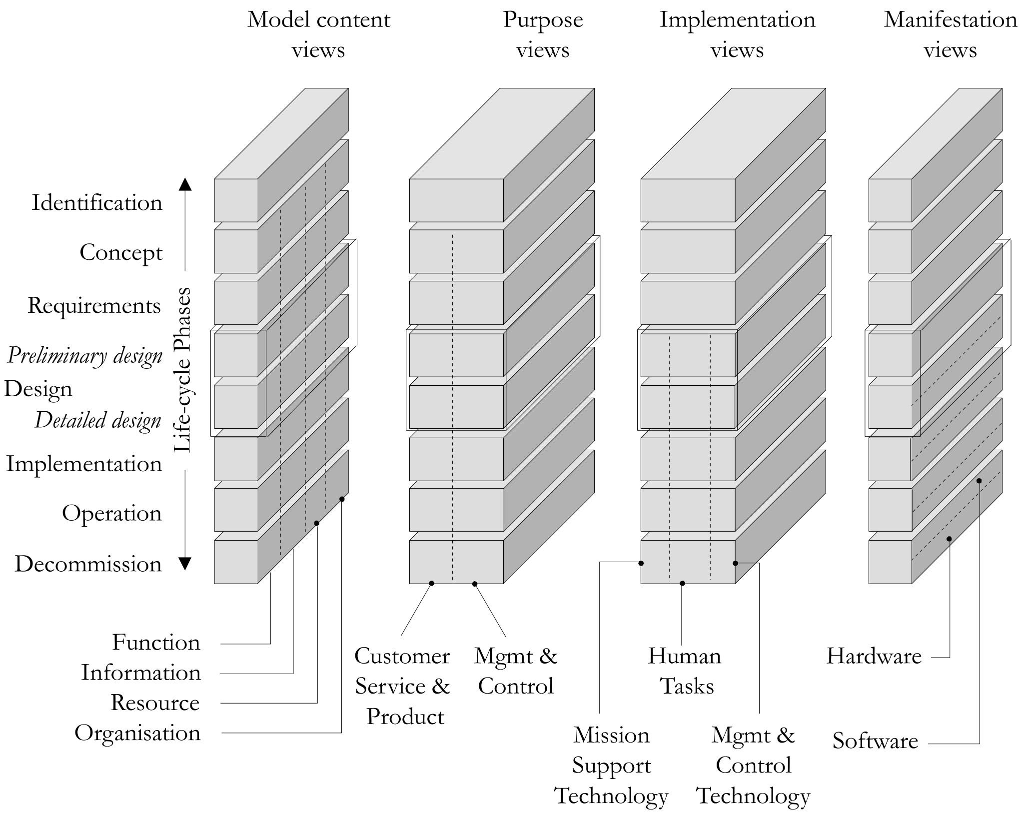

Figure 9 – The Modelling View concept. Four essential view types and their contents. Other modelling views may be defined if needed and supported by the engineering tools.

- *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 which may be done by machines (extent of automatability) and those which will be done by machines (extent of automation).

3.1.5.2.4 Entity Physical Manifestation Views

Again, two different views allow 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, that can be stored in a computer or in any other control device enabling the execution of operational task, a set of instructions for humans with defined skills, such that the instructions for the humans to perform a task that 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 that 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 machinery with given functionality.

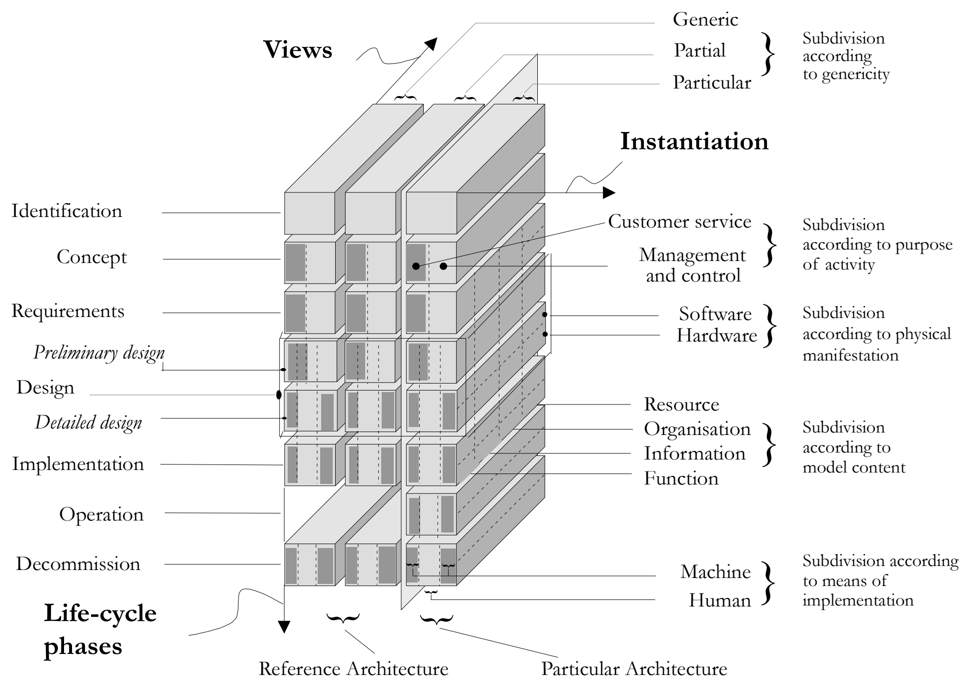

Figure 10 – GERA Modelling Framework with Modelling Views

Figure 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.

3.2 EEMs – Enterprise Engineering Methodologies

Enterprise engineering methodologies describe the processes of enterprise integration. 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 integration and will guide the user in the engineering tasks of enterprise modelling. Different methodologies may exist that 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 integration process itself is usually directed to a repetitive service- or manufacturing enterprise or a project enterprise. The methodology may be specifically oriented to the type of enterprise or entity under consideration.

Enterprise 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).

3.2.1 Human factor

The major part of a methodology is a structured approach that 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 Fig.11)

This action will separate the tasks of Mission Fulfilment and Management and Control as defined in the Requirements Analysis phase into three. Thus, the tasks or functions involved are assigned to the appropriate boxes that in turn define the automated information tasks that become the Management and Control Information Systems Architecture functions, and the automated manufacturing and service tasks that become the Mission Support 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 Management and Control Information System Architecture, the Human and Organisational Architecture, and the Mission Support 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.

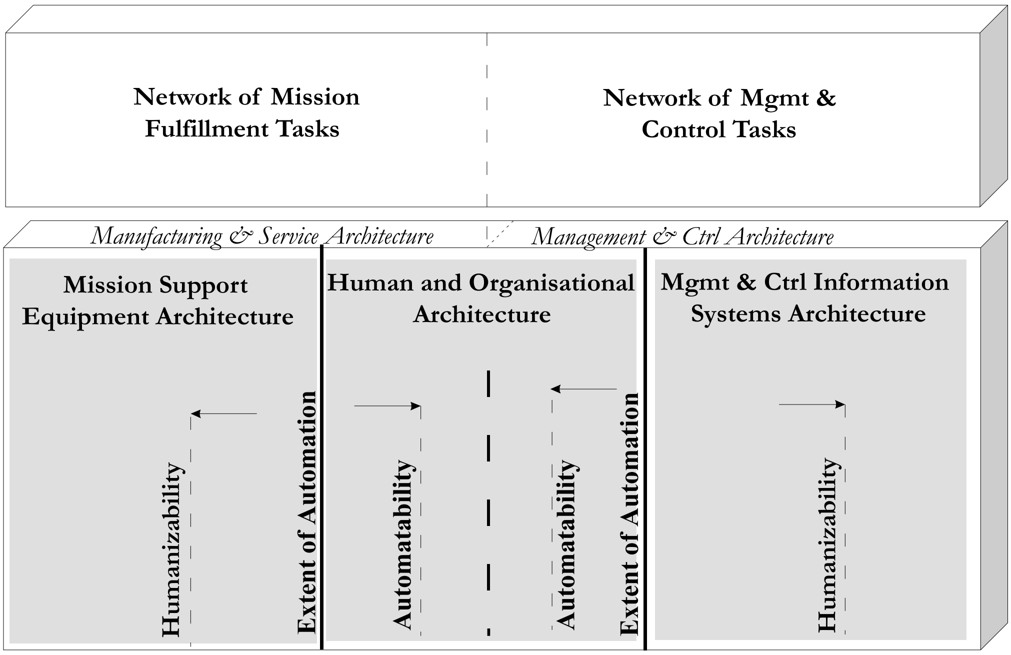

Figure 11 – Introduction of the Concepts of Automatability, Humanizability and Extent of Automation Lines to Define the Three Implementation Architectures

Still a third line, the Extent-of-Automation Line, can be drawn that 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 Management and Control Information System Architecture on the one side, and the boundary between the Human and Organisation Architecture and the Mission Support 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.

3.2.2 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 "Entity types in Enterprise Integration"). 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 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...), 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.

3.2.3 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 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.

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 3.1.4.2). 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 then the set of areas to be modelled, with one language suitable for more then one area.

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 8 and 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, or 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 that 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 [18], CIMOSA [8,14], GRAI/GIM [15], IEM [17] or the IDEF family of languages [16,8].Relevant Standardisation: ENV 12 204 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,). ISO DIS 14258 defines Rules and Guidelines for Enterprise Models.

3.4 GEMCs - Generic Enterprise Modelling Concepts

Generic Enterprise Modelling Concepts are the most generically used concepts and definitions of enterprise integration and modelling. 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 then one form of the above must be defined in a mutually consistent way;
- Those concepts that 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

3.4.1 Glossary

The terminology used in enterprise integration 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 that are defined in the semi-formal meta-models and / or formal ontologies.

3.4.2 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.

3.4.3 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.

[10] Meta-models are models about 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 define 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]

3.5 PEMs – Partial Enterprise Models

Partial enterprise models (reusable reference models) are models that 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 that capture some common part of a class of enterprises,
- Paradigmatic (reference or prototypical) models that 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 that capture the commonalities but leave out specific details. This type of model is of the 'fill-in-the-blank' type.

3.5.1 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.

3.5.2 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 3.1.5.2), 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. For example, 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 that 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 that ensure that the resulting models define a high-quality, business-process model as well as feasible system implementation.

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

3.5.3 Partial Technology Models

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.

3.5.3.1 Partial Models of IT systems

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

One important partial model commonly needed for enterprise integration is the one of integrating services (see chapter 3.1.4.1) that provide portability across heterogeneous environments. These services have to include communication, processing/execution control, 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 3.7).

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
- provide a shared design repository, or database, that 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 that 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 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 that 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) [18], FirstSTEP (CIMOSA), MOGO (IEM) [19], KBSI Tools [20], METIS, Processwise, 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.

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, 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 3.5.3.1).

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 that 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 that 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.

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 that provide the system specifications and identify the enterprise modules used in the system implementation.

4 References

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

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

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

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

[12] Weston, R.H. "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, RJ. 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] Spur, G.; Mertins, K.; Jochem, R.: 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).

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

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/TC310 "CIM Systems Architecture - Enterprise model execution and integration services - Evaluation report" CEN Report CR: 1831 :1995.
- CEN/TC310 "CIM Systems Architecture - Enterprise model execution and integration services - Statement of Requirements" CEN Report CR: 1832 :1995.

- CD 14258 " Industrial automation systems - Rules and guidelines for enterprise models" ISO TC184/SC5/WG1, 1996