15th Triennial World Congress, Barcelona, Spain

DESIGNING MANUFACTURING SYSTEMS: CONTRIBUTION TO THE DEVELOPMENT OF AN ENTERPRISE ENGINEERING METHODOLOGY (EEM) WITHIN THE FRAME OF GERAM

David Chen, Bruno Vallespir and Guy Doumeingts

LAP/GRAI
University Bordeaux I /ENSEIRB UMR CNRS 5131
351 Cours de la libération, 33405 Talence Cedex, France
E-mail: name@lap.u-bordeaux.fr

Abstract: Within the frame of the GERAM (Generalised Enterprise Reference Architecture and Methodology) developed by the IFAC/IFIP Task on Architecture for Enterprise integration, this paper presents a preliminary approach for the development of an Enterprise Engineering Methodology (EEM). The approach is currently limited to Entity requirements and Entity design phases of the GERA (Generic Enterprise Reference Architecture) system life cycle. It tentatively proposes a process to follow to identify user's requirements and to perform system design. The approach of the proposal is based on the GRAI methodology. Relevant concepts and principles from CIMOSA and PERA also take part in the approach. *Copyright ã 2002 IFAC*

Keywords: Design, Methodology, Manufacturing system, Architecture for enterprise integration.

1. INTRODUCTION

The IFIP/IFAC Task Force on Architectures for Enterprise Integration has elaborated a Generic Enterprise Reference Architecture and Methodologies (GERAM) (Bernus, 1997). It has been developed on the basis of some known approaches in this domain such as the Purdue Enterprise Reference Architecture (PERA) (Williams, 1996), the GRAI Integrated Methodology (GIM) (Doumeingts, *et al*., 1992, 1998), the Computer Integrated Manufacturing Open System Architecture (CIMOSA) (Amice, 1993). Based on the GERAM and with contributions from members of the Task Force (Vernadat, 1996), an international standard *IS 15704 'Requirements for Enterprise Reference Architecture and Methodologies'* has been elaborated. The approach proposed in this paper is developed within the frame of and consistent with GERAM.

The most important component of GERAM, called GERA (Generalised Enterprise Reference Architecture), introduces the basic concepts to be used in enterprise engineering and integration. It distinguishes between the methodologies for enterprise engineering (EEMs) and the modelling languages which are used by the methodologies to describe and model the structure, content and behaviour of the enterprise entities in question (Bernus, 1997).

An Enterprise Engineering Methodology describes the processes of enterprise integration along the all life cycle activities. The life cycle defined in GERA has seven phases (IS/15704, 1998): (1) <u>Entity identification</u> which identifies the contents of the particular entity under consideration in terms of its boundaries and its relation to its internal and external environments; (2) <u>Entity concept</u>: this phase contains the set of activities that are needed to develop the concepts of the underlying entity; (3) <u>Entity requirement</u> develops descriptions of operational requirements of the enterprise entity, its relevant processes and the collection of all their functional, behavioural, informational and capability needs; (4) Entity design consists in transforming requirements on specifications that allow to build the entity with all of its components that satisfy the entity requirements; (5) Entity implementation phase defines all those tasks which must be carried out to build or re-build (i.e. manifest) the entity; (6) Entity operation: the activities of the entity which are needed during its operation for producing the customers product or service which is its special mission along with all those tasks needed for monitoring, controlling, and evaluating the operation; (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.

Within the frame of the GERA system life cycle, this paper aims at contributing to the development of an EEM. It is generic and independent of a particular modelling language. At current stage of the development, the approach presented hereafter is limited to Entity Requirement and Entity Design phases of the life cycle. The main part of the proposal is based on the GRAI methodology. Concepts and principles from other known methodology and architecture (in particular CIMOSA and PERA) are also part of the approach. It focuses on defining what is to be done and in which sequence it is to be done rather than how it is to be done.

2. ENTITY REQUIREMENTS

Define precisely user's requirements is a very difficult task. We generally know how to define the main functions for the normal use of the product or a system, it is extremely difficult to express the functions that are linked to the context of the use of the product or the system because they are often implicit functions (Tichkiewitch, 1999). Because requirements can hardly be completely and consistently identified when design starts, design itself involves activities to identify, to revise and to refine design requirements (Grabowski, 1999). Design is an evolutionary process alternating from mapping of requirements to solution and refinement (Chen *et al*., 2002). Consequently, the process to follow from Entity requirements to Entity design is not straight forward. Interaction and iteration between the two phases would happen at each step.

2.1 Preliminary remark: existing system modelling

In the proposed approach, we considered that there is always a phase of modelling the existing system before design phases (see fig.1.). The reason is that there are very few real design projects in manufacturing. Most of time, a manufacturing system still exists and the issue is to make it evolve. Then, the design of a new system is a combination of existing system and requirements . The two main interests of modelling the existing system are (1) to understand the system and its constraints (internal or from the environment) and (2) to define all along the design what must be changed and what can be kept (see below).

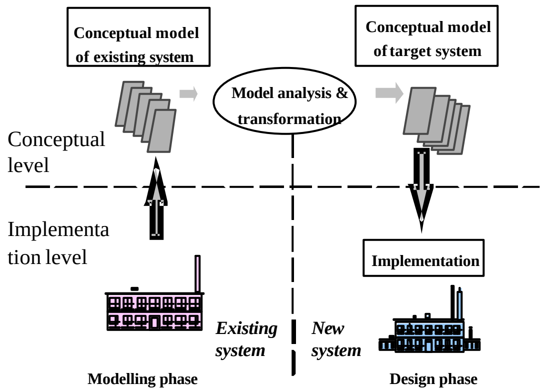

Fig. 1. Generic approach (GRAI methodology)

Existing system components identified at the implementation level are to be synthesised and abstracted into a set of technology independent models (example: function, decision, information,…models) at the conceptual level. Using some pre-defined rules, analysis can be done on these models in order to understand the real requirements for future system and to detect possible inconsistencies. The set of models that represent requirements of future system can be derived / transformed from the models of existing system.

2.2 Functions definition

This step aims at defining the main functions that the future system must provide to accomplish its mission. For example, a manufacturing shop might require functions such as turning, drilling, transporting,… These functions may be established hierarchically. They represent the functionality of the system and are static. Functions include not only manufacturing functions, but also information processing ones and management functions. Functional requirements are the most important ones because they express the finality of the system (why the system has been or has to be designed).

2.3 Processes elaboration

A process is a partially ordered set of activities. Elaborate a process is to define what the system must do in order to fulfil a or several functions. A process can be decomposed on sub-processes. The lowest level of decomposition contains activities

that will perform the work. Processes represent the behaviour of the system. They are the dynamic representation of the system.
A good way to start processes elaboration is to list all events from the environment of the system that it must react to. Examples of events are: reception of a customer order, machine break-down, shortage of a raw material, etc. Then, events identified here will trigger processes when they will be elaborated. Let us notice that some management processes such as production planning are triggered periodically without necessarily other outside events.

2.4 Decisions definition and structuring

This step consists in identifying all the levels of decision within each main management function so that decisions made within various functions are consistent in the sense that they contribute to the achievement of the global objective(s) of the company. As we know this is not always the case because the objectives of functions are often locally defined and might be incomplete even contradictory with global ones. That is why decision-making must be consistently structured from the upper level (decisions related to the control of the system as a whole) to the precise control of detailed operational activities. A good definition of decisions is mandatory in order to ensure that the system in operation will fulfil the functions defined above.

2.5 Information identification

For each activity defined previously, one must identify information/data that are necessary to perform the activity. Information requirements may be represented using some existing modelling languages such as Entity Relationship formalism, Object view etc. After having defined information per activity, it is very important to gather all of them into a consistent system. Then, information requirements will lead to the design of the information sub-system of the enterprise.

2.6 Capability description

At the entity requirements level, only required resource capability will be documented. This is to be done in relation with activities that represent the desired functionality. Defining resource capability is a preliminary mapping between functions and resources which is the main issue of entity design. For example, a "turning activity" might require a resource with the capability (characteristics of the resource) such as: speed of rotation, cutting precision, highest temperature allowed, etc. This step also allows to eliminate some unfeasible functional requirements (no available technology can answer the required function). It must be noticed that all the tasks that have been described right before may express requirements in term of capability.

2.7 Remarks

A structured approach of seven steps has been proposed for entity requirements phase. They are presented in a chronological order. Among them, identifying desired functions is the most crucial one. Because entity design mainly consists in mapping functions onto organs (resources). Functional requirements should be large and robust enough to cover various manufacturing needs (not only for manufacturing current families of products but also near future ones).
Functional requirements are different from the other ones because they are the most robust and steady. For example, manufacturing processes define operating procedures which represent the flow of manufacturing activities for a given type of products. When products evolve (change), processes will change accordingly. However functions remain the same (or change little).

2.8 Project organisation

During the requirement definition phase, quantity of information / knowledge will be collected and validated. This must be done in a structured way to avoid unnecessary iterations. Based on GRAI methodology, the project participants can be organised into following groups (see fig.2):

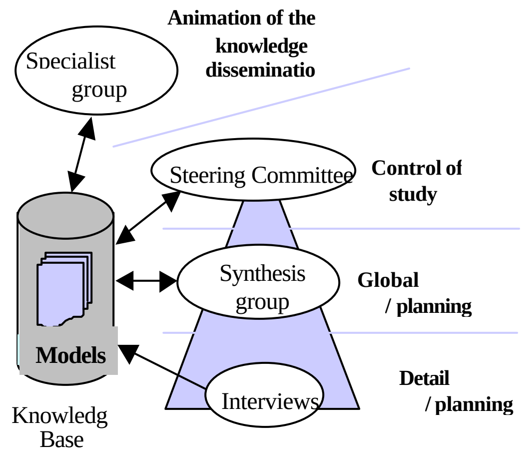

Fig. 2. Project participants (GRAI methodology)

- Steering committee: It is composed of the top-management members of the company. The role of this group is to express the objective of the study, and to orient the project.
- Synthesis group: This group is composed of main responsible people and users of the company. Its role is to ensure the follow-up of the project and to check the results at the various steps.
- Specialist group: It is a group of experts on enterprise architecture and methodology. Its

role is to advise the synthesis group, and to build various models.

- Interviews: Group of company people to be interviewed by specialists. They provide information needed by the other groups.

Two approaches are used to capture the information and knowledge: Top-down and Bottom-up.

- The top-down approach is conducted by the synthesis group by organising meetings where global models are presented and validated by main responsible people of the company.
- The bottom-up approach is carried out by the specialist group by organising interviews with users of the future system to elaborate detailed models.

The confrontation between results obtained from top-down and bottom-up approaches allows to detect some potential inconsistencies of the system and to better define requirements.

3. ENTITY DESIGN

Entity design phase follows entity requirements. Entity design is split into two sub-phases: preliminary design and detail design. This paper focuses on the preliminary design.

Preliminary design consists in translating requirements in design specifications which are intermediate solution that will be further specified by detail design. Generally speaking, design aims at transforming required functions, processes, decisions, information and capabilities into specifications of three domains: Business, IT and Organisation. In other words, it consists in elaborating the specifications on Manufacturing technology components (machines, robots, human operators,…), Information Technology components (hardware and software) and organisation structure (decision making, assignment of activities to people and services of the company including the notions of responsibility and authority).

At the preliminary design phase, the design specifications will be given at the level of solution types without necessarily choosing (deciding) the specific components. For example, one may decide to implement a NC type cutting machine and specify some characteristics (speed, dimension, precision,…). This specification will be further detailed (if necessary) at the detail design phase in order to select a commercial machine available on the market (Chen, *et al*., 2002).

3.1 Functions distribution and global assignment

After having defined functional requirements in entity requirement phase, it is necessary to specify distributed sub-systems / functions. It means to decide what function will actually be implemented and operates where. For example the functions F1.2, F1.1.1 and F1.1.2 are regrouped, they are to be implemented in site A. Function 1.3 will be implemented in site B. In some cases, more detailed functional decomposition will be needed if part of a function is to be distributed in site A, and another part is distributed in site B. Sometimes it is also necessary to aggregate several functions of the same nature into one if these functions are to be implemented in the same site. Distributed functions lead to a distributed information system. This may increase operational cost, generate delay because of information transmission and need supplementary co-ordination activities. This task supposes that the sites have been defined before the project or remain the same. In this case, they have been identified while the modelling of existing system.

3.2 Level of automation definition

Functions identified previously can be either performed by human or by automated / computerised equipment. Therefore one of the design decisions consists in determining the level of automation. For a given function, the designer must compare the actual technology available on the market with human abilities in terms of speed of response, physical strength, working conditions etc. Automation option is limited by the fact that many tasks and functions require human innovation, etc. and cannot be automated with present available technology (Williams, 1994). This consideration could also be based on economic criteria, i.e. the cost would be too high compared with the cost of humans. The design decision should be a balanced solution between the limit of automation and the limit of human capability.

3.3 Resources definition and functions assignment

According to the level of automation chosen and the capabilities description entity requirement), a manufacturing function will be assigned to either a human resource or a machine resource. At this stage, only resource types are to be determined. Decision on choosing specific commercial components will be taken after the detail design. For example, to assign a resource type to a function, one may prefer to use a NC machining centre having the three functions i.e. turning, drilling and milling instead of implementing three individual NC machines, each performing one function. Once the resource type decided, it is also necessary to complete and refine the description of that resource (characteristics of the resource) in order to choose a commercial component later.

Mapping functions to organs is also concerned with the Information Technology related resources. Generally speaking, there are three categories: (1) data storage devices (including hard disk, CD and optical record facilities); (2) data processing

devices (PC, mainframe,…); (3) communication and data transmission devices (for example, Local Area Network,…). As indicated before, at the preliminary design stage, only solution principles need to be determined. For example, one can choose decentralised PCs interconnected via a local network instead of implementing one mainframe computer with distributed terminals.
Another example is to determine the type of the data storage device without considering a specific commercially available system. For example, it could be manual or computerised. If it is computerised, the centralised solution may be chosen instead of distributed implementation. Moreover, database itself can be relational type or object-oriented, etc..

Because of the cost of acquisition of new resources, it is always better to try to assign functions to existing resources in the extend that their capability makes this possible. Existing resources have been described while the modelling of existing system.

3.4 Specify distributed information

Like functions, information/data also needs to be distributed where it is needed for the execution of functions. For example, if the modelling formalism used to represent information requirements is entity/relationship model, data entities are to be regrouped and re-arranged according to the user's use of these data entities (i.e. data entities will be grouped or aggregated to support distributed functions). External schema will be elaborated to specify interfaces with end-users.
As we have discussed before, distributed implementation of functions leads to a distributed information system. When deciding a such solution, several factors should be taken into account:

- Distributed information storage requires that one data has only one instance. However, end users for the reason of convenience often duplicate information and this raises the problem of updating information;
- Distributed system often generates more important information exchange than centralised system. This will increase the cost (cost of network, cost of maintenance etc.);
- Information exchange between sites may have delay. Proportionally speaking, this delay may become non-negligible for some types of applications.

3.5 Define organisation

Once resources (type) identified, it is necessary to re-group them in order to build an organisation structure. This organisation structure has two representations. One is the resource hierarchy in terms of for example factory, shops, sections, cells, workstations. The other is management hierarchy with corresponding responsibility and authority established with respect to resource hierarchy, for example the organisation chart. When regrouping resources, several known solution principles can be used as references. For example one may prefer group technology based organisation (a family of products will be entirely manufactured within one cell) instead of grouping machines according to their types. Another example is to organise resources in a Job-shop or a Flow-shop.

This step is also concerned with defining necessary responsibility and authority with respect to function/process, information and resources. It means to precise who is responsible for and authorised to create, use, update, modify and maintain functions, processes, activities, information and resources identified previously.

Even if the direct cost of implementing a new organisation is few, it must be kept in mind that it is always a hard and traumatising task for the company. That is why it is important to start by analysing in what extend the existing organisation might be kept. The existing organisation has been described while the modelling of existing system.

3.6 Mapping decisions onto organisation

Decisions to control a manufacturing system need to be structured and mapped onto the enterprise organisation for its implementation. At the entity requirements phase, decisions to be taken within the main management functions have been identified at the three levels (long term, medium term and short term). The design decision here is to assign resources (services) to these decisions. There are several possible cases: (1) one decision is taken by one service; (2) one decision is taken jointly by several services; (3) several decisions are taken by one service. Another issue is to ensure that each decision maker has at his disposal: (1) <u>objectives of decision</u> that are consistent to the global ones, (2) <u>variables of decision</u> upon which he/she can take choice, (3) <u>constraints defining his space of decision</u>, and (4) <u>criteria</u> to evaluate the quality of decision.

3.7 Specify interfaces

One of the issues is to design interfaces between various components of the system. Components are of three types: (1) Human type (operator and manager), (2) Machine type (including sensors, automated storage and transport sub-systems and modules), (3) Computer type (including applications and databases). Consequently, the interfaces are interconnections between these resources. The interconnections are electronic and/or mechanical which permit two or more physical (or human and organisational or both) modules to carry out the information and material and energy transfer functions of the two or more functional modules which are interconnected.

3.8 Economic evaluation

Economic evaluation could be carried out during and after the entity design. During design, each time an important design choice is taken (for example to distribute a function), costs should be compared with potential benefits to be generated by implementing that solution. The economic evaluation to be performed at the end of the entity design phase consists in checking if the cost of proposed design fits within the frame of the budget agreed beforehand. This allows to go back easily to previous design decisions event to modify requirements before starting the detail design.

3.9 Remarks

Designing manufacturing systems is not only a technical issue, various factors and constraints may influence design decisions, such as financial implication, risk, compliance to standards, external restrictions, flexibility and feasibility etc..
The assignment of resources to functions is one of the most important issue of the preliminary design. Because resources constitute main organs of the system structure. This structure will determine the main functionality and behaviour of the designed manufacturing system.
Generally speaking, when mapping functions onto technology or human implemented sub-systems, it might appear necessary to reconsider functional decomposition. For example, some functions of the same nature may need to be aggregated into one sub-system that is to be automated, some others be aggregated into another sub-system to be implemented by humans.

4. CONCLUSIONS

This paper has tentatively presented a structured approach aiming at contributing to the development of an Enterprise Engineering Methodology (EEM). It has been drafted within the frame of GERAM to show one possible way to design a particular manufacturing system using the concepts proposed by GERA. The approach is based on GRAI methodology and consistent to CIMOSA and PERA. It is currently limited to the two phases of the life cycle (entity requirement and entity design) and focuses mainly on technical issues. Consequently it should be considered as a basis and needs to be further developed.

Future works would be concerned with:

- the refinement of the proposed process and possibly the use of Suh's independent axiom (Suh, 1990) to better structure various steps.
- the development of the set of basic design principles and solution types to use for the preliminary design.
- the extension of the approach to the whole life cycle phases to elaborate a precise roadmap to follow from entity identification to entity decommissioning.
- the development of a case study to illustrate step by step how to use the methodology when studying a particular enterprise system.
- the consideration of other aspects such as project management (including budgeting, planning and follow-up), economic evaluation, social and human factors etc..

REFERENCES

AMICE (1993). *CIMOSA: Open System Architecture for CIM*, Springer-Verlag, Berlin.

Bernus, P. and Nemes, L. (1997). The contribution of the Generalised Enterprise Reference Architecture to Consensus in the area of Enterprise Integration. In Proceedings of ICEIMT'97, Editors Kosank, K. and Nell, J.G.

Chen, D., Vallespir, B. and Doumeingts, G. (2002). Translating requirements in design specifications, *In Handbook for enterprise architecture*, Editors: Bernus, P. Nemes, L. and Schmidt, G. Springer Verlag, To appear.

Doumeingts, G., Vallespir, B., Zanettin and M., Chen, D. (1992). GIM: GRAI Integrated Methodology for designing CIM systems. *GRAI/LAP, University Bordeaux I*, version 1.0.

Doumeingts, G., Vallespir, B. and Chen, D. (1998). Decision modelling GRAI grid, Chapter in: *Handbook on architecture for Information Systems* (Peter Bernus, Kai Mertins, Gunter Schmidt. (Ed)), Springer.

Grabowski, H., Lossack, R. and El-Mejjbri, E. (1999). Towards A Universal Design Theory, *In Proceedings of the CIRP International Design Seminar*, University of Twente, The Netherlands, March 24-26, 1999.

IFAC-IFIP Task Force (1997). GERAM: Generalised Enterprise Reference Architecture and Methodology, *ISOTC184 /SC5/WG1, N398*.

IS 15704 (1998). Requirements for Enterprise Reference Architecture and Methodologies. *ISO TC 184/SC5/WG1*.

Suh, N. P. (1990). *The Principles of Design*. Edition Oxford University Press.

Tichkiewitch, S. and Roucoules, L. (1999). *Methodology for innovative design*, CIRP International Design Seminar, University of Twente, The Netherlands, 24-26 March 1999.

Vernadat, F. B. (1996). Enterprise Modelling and Integration – Principles and Applications. *Chapman & Hall*, ISBN 0-412-60550-3.

Williams, T.J. (1994). The Purdue Enterprise Reference Architecture, In *Computers In Industry*, vol. 24. 1994.