The M*-object methodology for information system design in CIM environments : the organisation analysis phase

Antonio Di Leva, Piercarlo Giolito, François Vernadat

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The M-object methodology for information system design in CIM environments : the organisation analysis phase*

Antonio DI LEVA
Piercarlo GIOLITO - François VERNADAT

N° 1918
Mai 1993

PROGRAMME 5

Traitement du signal,
automatique et
productique

THE M*-OBJECT METHODOLOGY FOR INFORMATION SYSTEM DESIGN IN CIM ENVIRONMENTS: THE ORGANISATION ANALYSIS PHASE

LA METHODOLOGIE M*-OBJECT POUR LA CONCEPTION DES SYSTEMES D'INFORMATION EN ENVIRONNEMENTS DE CIM : LA PHASE D'ANALYSE DE L'ORGANISATION

Antonio DI LEVA*, Piercarlo GIOLITO*, François VERNADAT**

* Università di Torino, Dipartimento di Informatica, corso Svizzera 185, I-10149 Torino, Italy

** INRIA-Lorraine/CESCOM, 4 rue Marconi, F-57070 Metz, France

Résumé
M*-OBJECT est une méthodologie pour l'analyse, la conception et l'implantation des systèmes d'information développés dans les environnements CIM. Elle est basée sur une approche orientée objets et elle couvre l'analyse approfondie des aspects statiques et dynamiques des informations.
M*-OBJECT comprend trois phases principales: l'analyse de l'organisation, l'analyse conceptuelle et l'analyse de l'implantation. La phase d'analyse de l'organisation produit un ensemble de besoins structurés. La phase d'analyse conceptuelle fournit une spécification formelle exécutable du système d'information. La phase d'analyse de l'implantation fournit la description des bases de données nécessaires à l'implantation du système d'information du système CIM. A chaque phase, un formalisme dédié et un ensemble de directives sont fournis qui peuvent être informatisés.
Ce rapport concerne la phase d'analyse de l'organisation pour laquelle un modèle d'organisation basé sur les concepts d'agent, d'évènement, de fonction (processus et activité) et de composant est proposé.

Mots-clés
CIM, Conception de systèmes d'information, Analyse d'organisation, Méthodologie M*

Abstract
M*-OBJECT is a methodology for information system analysis, design and implementation developed for CIM environments. It is based on an object-oriented approach and it covers in-depth analysis of static and dynamic aspects of the CIM information system.
M*-OBJECT is made of three major phases: organisation analysis, conceptual design and implementation design. The organisation analysis phase provides a structured set of requirements. The conceptual design phase provides executable formal specifications of the information system. The implementation design phase provides the implementation description of necessary databases. Each phase is supported by a dedicated model and set of guidelines which can be computerised.
This article focuses on the organisation analysis phase for which an organisation model based on the concepts of agent, event, function (process and activity) and component is proposed.

Keywords
CIM, Information system design, Organisation analysis, M* methodology

1. Introduction

Computer-Integrated Manufacturing (CIM) is the efficient use of Information Technology and Manufacturing Technology to improve productivity and efficiency of modern manufacturing enterprises. Obviously, the design of the information system of such manufacturing systems is a crucial endeavour in implementing a successful CIM environment. Usually, the information system is implemented in the form of one or more heterogeneous database systems.
A general methodological approach for the development, implementation and operational use of production and CIM databases involves the following steps:

1. specification of the CIM system (definition of a strategic plan);
2. enterprise modelling and analysis (using enterprise analysis methods such as IDEF [1,2], CIMOSA [3] or the Purdue Reference Model [4]);
3. database design committee set up;
4. information modelling and conceptual database design;
5. selection of the appropriate physical database management systems (DBMS);
6. logical and physical design, and implementation of the database;
7. application program development;
8. database operational use and maintenance.

The fist step is performed by top management and consists of an explicit statement of what has to be automated in the enterprise. The second step provides a clear understanding of the actual state of the enterprise (or part of it) and what needs to be improved and automated according to the strategic plan produced by the top management. Thus, the scope of application (or area of concern) of the necessary databases can be identified. Once this step is completed and approval for the automation project has been issued, a database committee is formed. This is a group of people (from the company or from outside), the mission of which is to analyse the information system and to define the database system. The next step is information modelling and conceptual design of the database structures by the database committee. Then, one or more commercial DBMSs can be selected according to the specific requirements of the application project. The logical and physical design of the databases for the particular DBMSs can now take place and the databases can be implemented with these systems. Application programs can be developed and coded by application programmers. Finally, operational use of the databases can be started when application programs are completed and debugged.
An information system design methodology consists of models (and the related languages) used to describe the part of the real-world under analysis (e.g. an enterprise), and methods, i.e. design startegies to build up the real-world description. According to a survey [5], methodologies were first based on hierarchical decomposition of business activities and on the functional description of interactions between the resulting enterprise components. An exemplification of this approach is the SADT methodology [6] and related methodologies such as PSL/PSA [7] and JSD [8].
Such types of methodologies, in our opinion, are still useful for the analysis phase but suffer from a severe lack of formal methods to specify, in the design phase, all the static, dynamic, and behavioural aspects of the object enterprise. In fact, recent development of information system design methodologies focus on the integration in the design phase of concepts and constructs to deal with all these aspects. Such methodologies include REMORA [9], TAXIS [10], and TEMPORA [11].
Our work in this field started with M* [12,13], which is an information system design methodology developed for CIM environments, and complying with the general framework previously outlined. M* makes use of an extended entity-relationship approach [14] to describe data structures in the conceptual design phase. Currently, the object-oriented approach is gaining tremendous interest in the CIM community because of its modelling power and expressiveness, its suitability for building ontologies in engineering and manufacturing domains, its capabilities for model reusability and schema evolution. It was therefore necessary to adapt the M* methodology to this new paradigm, giving birth to the M*-OBJECT methodology discussed in this paper.

The aim of the M*-OBJECT methodology is to provide engineers with methods and tools for information system analysis and design as well as development of database applications in production and CIM environments.
Therefore, according to the general framework given above, the M*-OBJECT methodology covers steps 2, 3, 4, and 6, i.e. the rough analysis of a manufacturing organisation, the modelling of its information system, and the design of the related database environments which will support the automated operations of the organisation. Also, step 7 of the framework is partially covered.
The paper provides a general presentation of the M*-OBJECT methodology and a discussion of the Organisation Analysis phase (The Conceptual and Implementation Design phases will be discussed in a subsequent paper [15]). More specifically, Section 2 presents the architecture of the methodology. Section 3 describes a simple example that will be used to illustrate the methodological phases. Sections 4 to 8 describe the Organisation Analysis phase and the organisation model, and Section 9 states our conclusions.

2. The M*-OBJECT General Architecture

The M*-OBJECT methodology consists of three main phases: (a) Organisation Analysis, (b) Conceptual Design, and (c) Implementation Design.
The aim of the Organisation Analysis phase is to analyse precisely the current status of the organisation and to define a new, more productive or more efficient organisation structure for the production system under analysis (or object enterprise). The output of this phase is a description of the *system environment*, i.e. the framework within which the information system design and implementation will take place, and the specification of *system requirements*, i.e. a high level, user-oriented, description of the components of the object enterprise that will be automated.
The Conceptual Design phase is used to analyse the organisational description (i.e. the system environment and the system requirements) and to construct a conceptual specification, the *conceptual schema*, of the object enterprise. The conceptual schema is an executable specification of both static and dynamic enterprise features. Static information refers to the structure of data and to the integrity constraints that data must satisfy. Dynamic information refers to the organisation behaviour, i.e. the operations that must be performed on data and to the causal relationships existing among them.
The conceptual schema consists of a database schema and a set of processes which run on the database. The Implementation Design phase translates the conceptual schema into the information system specification (or *implementation schema*) processable by the target DBMS.
The first two phases of the M*-OBJECT methodology are independent of the specific DBMS and software environment that will be used for the application development. Conversely, the third phase is strictly related to the target database system. Since the major aim of the M*-OBJECT methodology is information system design and not organisation system design, it is supposed that other "ad hoc" methodologies (see, for instance, [16]) have been applied to deal with important aspects of the organisation analysis such as the feasibility study and the cost-benefit analysis.
The overall architecture of the M*-OBJECT methodology is illustrated in Fig. 1 .
The M*-OBJECT methodology is supported by an integrated set of models that allow the description of the object enterprise at three different levels, which correspond to the different phases of the methodology: (a) Organisation Level or management level for which a clear and accurate representation of the enterprise behaviour is required, (b) Conceptual Level or system level which provides a common view, a bridge, between managers and engineers on one hand and computer specialists on the other hand, and (c) Implementation Level or application development level only of interest to computer specialists and application programmers. In this paper, the organisation model will be described and illustrated by a simple manufacturing case study.

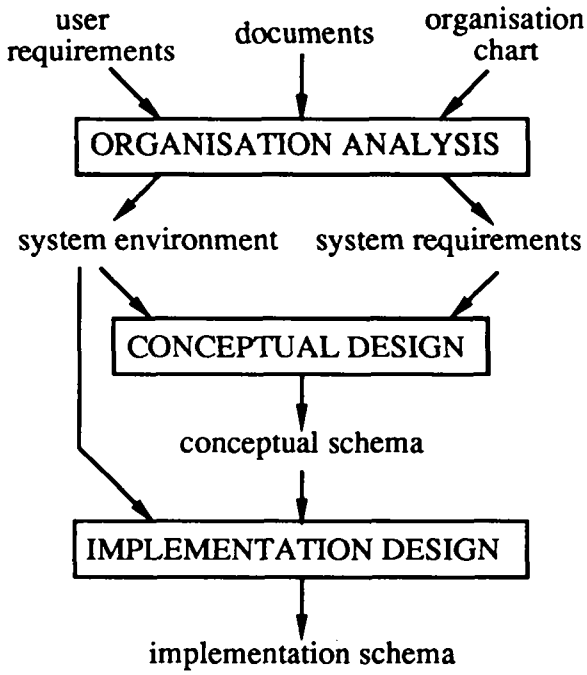


Fig. 1: The Overall Architecture of the M*-OBJECT Methodology

3. The Case Study

Let us consider a hypothetical company, called ABC, which will be used for illustration of the various phases of the M*-OBJECT methodology.
The ABC company is a manufacturing enterprise having a Manufacturing Resource Planning function (MRP-II), a Product Design and Process Planning function (CAD/CAPP) and a Shop Floor Control function (SFC) where manufacturing operations are performed using a flexible manufacturing cell (FMC). The FMC is composed of a number of machine-tools (including NC-machines, robots, measuring machines, ...), some clamping devices and pallets, cutting tools, measuring tools and devices. The aim of the FMC is to manufacture fabrication lots according to some predefined schedules. A fabrication lot is a batch of similar parts to be produced. A schedule is a list of manufacturing orders indicating for each order, the size of the lot, its due date, and possibly its priority. Each occurrence of a part of the same lot is a workpiece made of some material. The workpieces are manufactured according to a process plan. The process plan indicates the sequence of steps (machining operations, part/material handling operations, control operations) to be performed for a given part. It also indicates the machine to be used with the tool type for each type of operation. Therefore, it is necessary to store all the kinds of operations that each machine can perform and the types of tools which can be used for each operation type. Finally, for a given part, standard times per possible operation type (such as set-up time, labour time, queue time) must be recorded for subsequent analysis. In this case study, NC programs are not considered as real objects but just as manufacturing resources. Thus, only the references to the files containing them are stored in the database.

4. The Organisation Analysis Phase

In the Organisation Analysis phase, we use a simple, 3-level, organisational architecture to describe enterprises structured as follows. An object *enterprise* (or object organisation) is a real-world system (company, production system or plant, set of activities, ...) subject to be the focus of the analysis. An *enterprise environment* (environment for short) is a subset of the enterprise made of users and functions sharing a common view of the information system of the enterprise. An environment consists of *organisation-units* which, in turn, can be decomposed into *work-centres*. Such hierarchical

organisational elements are defined in terms of system objectives, system constraints and clustering of system functionalities at different organisation levels, so that there exists a high degree of cohesion within the same component and tight coupling between components.
Despite the fact that functional decomposition or system breakdown can be considered as a powerful approach for analysis of complex systems [17,18], there is no general approach or well-defined criteria to guide the decomposition process. Current methodologies for system decomposition usually provide decomposition "rules of thumb" only. Formal approaches for information systems have been presented in the literature recently (see, e.g. [19]).
A common problem faced by designers concerning enterprise decomposition is the specification of boundaries between components. At work-centre level, a user *activity* can be defined as a homogeneous set of *actions*, i.e. elementary steps, which can be executed without interruptions and independently of other activities [16]. Activities are then clustered into *processes*, which can be defined as a coordinated set of activities which accomplish an objective of an organisation-unit. Several processes can be coordinated to form an environment *function*, i.e. a more generic task.
The Organisation Analysis phase of the M*-OBJECT methodology is aimed at supporting a fine structural analysis of the enterprise. Its first task, the AS-IS Analysis, consists of a top-down analysis (from management to production functions) followed by a bottom-up analysis (starting with the production functions) and involving the users of the system to be modelled. Then, a TO-BE Analysis is performed which comprises identification of problems of the production system and specification of the new automated system. Finally, an Environment Specification task is performed in order to define the limits and scope of the database application (or system environment) to be designed. The overall architecture of the Organisation Analysis phase is shown in Fig. 2.

5. The Organisation Model

The organisation model is used to describe the various functional units of the object enterprise at the three different levels of details (environments, organisation-units, and work-centres).
The basic idea in the M*-OBJECT Organisation Analysis phase is that the real-world can be described using the following concepts: agent, component, event, time, and, depending on the abstraction level, function, process, and activity. More specifically, analysis tasks are carried out using the M*-OBJECT meta-model described in Fig. 3.

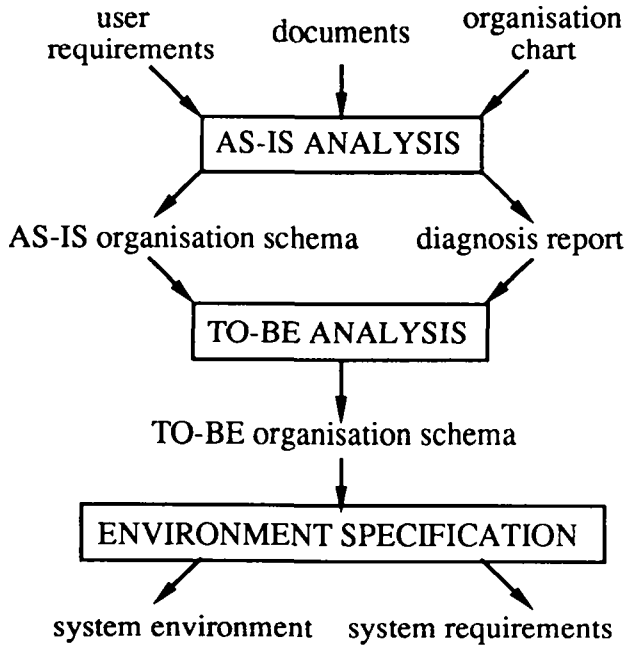


Fig. 2: The Organisation Analysis Phase

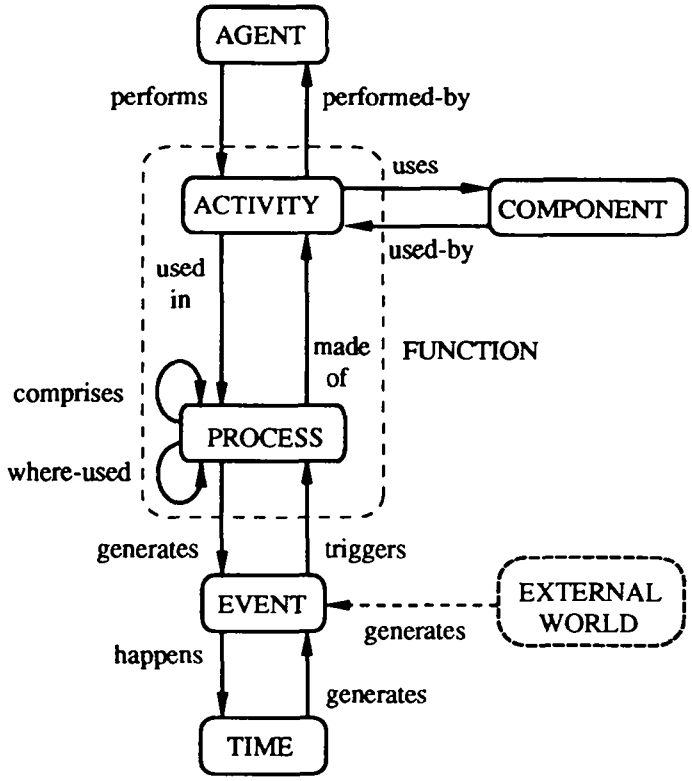


Fig. 3: The M*-OBJECT Meta-Model

According to the M*-OBJECT meta-model, the real-world is made of *components* that are handled by *functions* at the environment level (rough requirements analysis), and by *processes* and *activities* at organisation-unit and work-centre levels (detailed requirements analysis).
Components describe things of interest for the organisation, such as products, materials, support devices, tools, documents, files. Functions are performed by *agents* and can be triggered by *events* which correspond to specific state changes of either the object system or the external world in which the system is embedded. An agent may correspond to individual persons, working teams, active or autonomous technical devices, applications, and so on.
At the lowest levels of details of specification, functions are decomposed into processes. A process can employ ("comprises") or be employed by ("where-used") other processes, and be "made-of" activities, which are performed by agents. Processes are triggered by events, which express dynamic dependencies among processes and happen at a given moment in *time*. Events can either be generated by the external world in which the object organisation is embedded (for instance, an order has arrived), within other processes of the organisation, or as a consequence of a time condition (e.g. every day at 5 p.m., at the end of the year, ...). An event may trigger the execution of one or more different processes.
Activities imply a change of state in the organisation domain. Usually, activities depend on each other in a complex dynamic way; for instance, it has no sense executing a certain activity A before one or more other activities B, C, ... are terminated if their outputs provide input components to A. Causal relationships between activities, i.e. their flow of control, are described by the process they belong to.
Both the AS-IS and TO-BE analyses are specified by means of a generic model, called *organisation model*. Basic concepts of the organisation model are *object views* and *functions*. Object views are abstract units which hold information about system components, agents, and events. Functions are abstract units which describe, at different levels of detail, organisation functions, processes, and activities (i.e. things to be done) of the enterprise with their inputs and outputs (defined as object views).

An organisation net is a bipartite graph N = (**O**, **F**; **A**), where **O** and **F** are two disjoint sets of object views and functions respectively, and **A** is a set of directed arcs connecting object views to functions and vice versa. An organisation net is thus a descriptive causal model of a part of the enterprise (usually, it describes an organisation process). It is made of object views and functions in which functions indicate how object views are consumed, used or produced. The flow of objects determines precedence relations between functional components, but no explicit time dimension is introduced. The model has then the same expressive power than data flow models [20] or channel/agency nets [21].

Object views emphasized in M*-OBJECT with respect to CIM are:

- *messages*: represent real-world events or requests to execute a function as well as preconditions (input messages) and postconditions (output messages) to the execution of functions;
- *information*: represent information objects used/produced by functions. Four sub-classes have been defined: data for information items, files for structured sets of data items, forms for structured documents or computer screens and text for free-format text;
- *materials*: represent physical matters used/produced by functions (e.g. raw materials, parts, products, ...);
- *resources*: represent physical means used to carry out functions (e.g. tools, fixtures, machines, people, ...).

The organisation model is also used to describe the intrinsic behaviour of system components. This refers to the set of activities which are involved in the change of the state of a single component. At the organisation level, we are interested in so-called life cycle nets which are state-transition diagrams used to specify the possible sequences of execution of activities for a given component. In such nets, object views describe well-defined states of the component (for instance, for a part we can have part ready, part mounted on pallet, part finished, etc.), activities represent the evolution steps of the component from a pre-state to a post-state (e.g. mount part on pallet, machine part, etc.). Life cycle nets are applied in the organisation analysis phase to construct and verify the organisation nets.

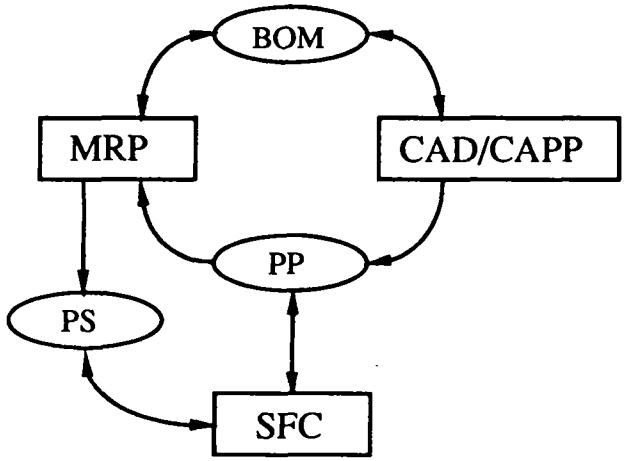
6. The AS-IS Analysis Task

The aim of the AS-IS Analysis task is to provide managers and engineers with an accurate model of the enterprise as it stands from which they can make a good assessment of its current status.

In the AS-IS Analysis, we distinguish two steps: top-down analysis (going from management downto production functions) and bottom-up analysis and diagnosis (starting from the production functions).

The goal of the top-down analysis step is to quickly understand the overall structure of the object enterprise, i.e. to discover the hierarchical structure of its decision system, identify the various levels of decision-making, and identify the decision-making centres. This step is crucial when the analyst comes from outside the enterprise to be analysed (in general, it is recommended that the analysis team should not be too familiar with the enterprise activities to be analysed to avoid any biais).

Usually, top-down analysis is performed on the basis of meetings and interviews involving top and middle managers and decision-makers. Discussions concern the general organisation of the enterprise, the size of the enterprise, the financial situation, the general problems faced by the enterprise, and so on. One meeting can be used to present the analysis tools and approach to managers. Other inputs to this task are the enterprise organisation chart (if it exists) and documentation brochures. These documents and interviews provide information about the gross organisational structure of the enterprise in terms of its objectives, environments (customers, suppliers, sub-contractors), departments, managers, plants, kinds of services produced, and so on. Enterprise elements are then classified as environments, organisation-units, work-centres as defined previously, and analysed in order to identify all major functions, as illustrated in Fig. 4 for the company ABC. In Fig. 4, functions have the same name than the environments they belong to. Object views are represented with ovals, functions with rectangular boxes and arcs represent major object flows. Note that it is also possible to use different shapes of boxes to distinguish various types of object views (data, messages, forms, files, materials, ...) [12]. In this paper, we only use one type of boxes.



MRP : Materials Requirements Planning

SFC : Shoop Floor Control

CAD/CAPP : Computer Aided Design and Process Planning

BOM : Bill of Material

PP : Process Plan

PS : Production Schedule

Fig. 4: The General Organisation Net for the Company ABC

The functional (structured) decomposition (or top-down refinement) is used to face the problem of managing real-world complexity: in most real-world situations, it is not possible to obtain directly the overall representation of the organisation at a given level of detail. According to the M*-OBJECT meta-model, for each refinement step functions and processes are modelled at the required level of detail (as decided by users). More specifically, for each function, the modelling starts by taking into account external events and the processes and activities which are triggered by them. These activities may generate new events, use new components and so on. In this way, an "outside-in" strategy seems to be the most natural way to perform the modelling step, starting from interfaces (external events exchanged with the external world). This step can procede "forward" or "backward" if input or output events are considered first, respectively.

The goals of the bottom-up analysis and diagnosis step are: (a) to validate the top-down analysis, (b) to finely model each function identified in the organisation net at a given level of detail, and (c) to make a diagnosis of each process in order to evaluate its operations.

A component life cycle is built for the most relevant components used by each function. A life cycle describes how a component can evolve in the organisation, i.e. it models all the possible sequences of activities which can be applied to the component and how its state is modified. Usually, life cycles represent a stable description of the intrinsic behaviour of typical components of a given enterprise. Thus, they can be stored in a life cycle library and later reused for several design projects. For instance, the Production Schedule (PS) and the Process Plan (PP) are important objects in CIM environments and their life cycles are given by Fig. 5 for the ABC company.

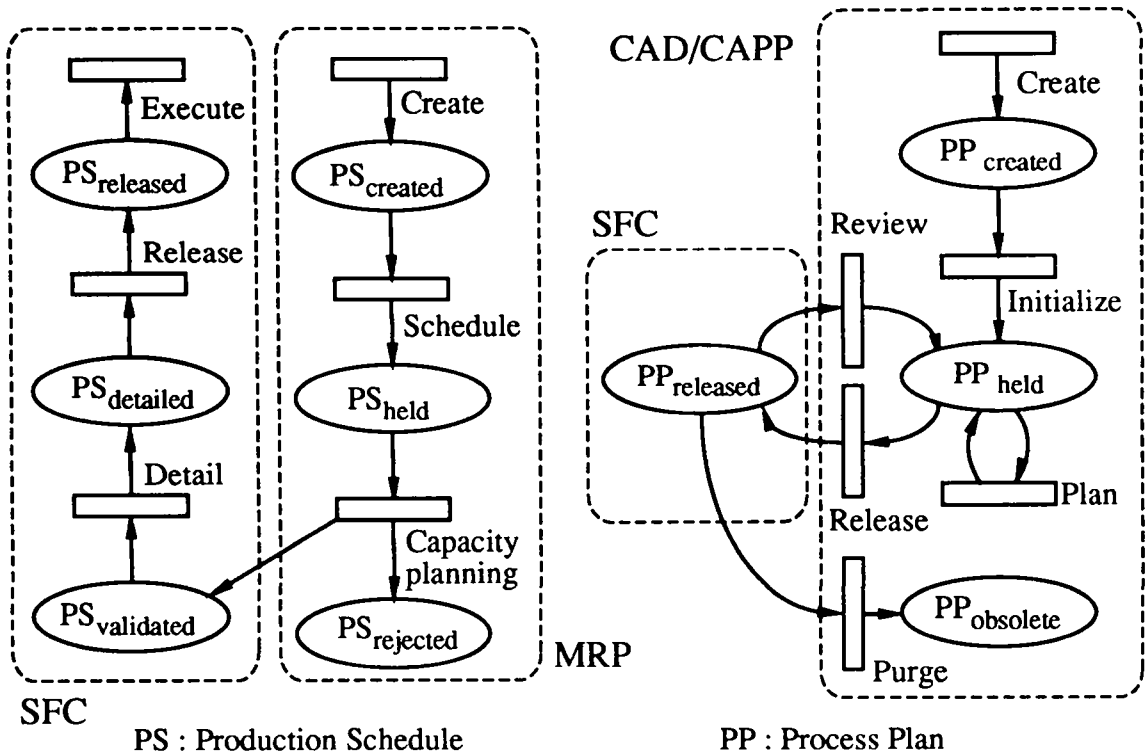


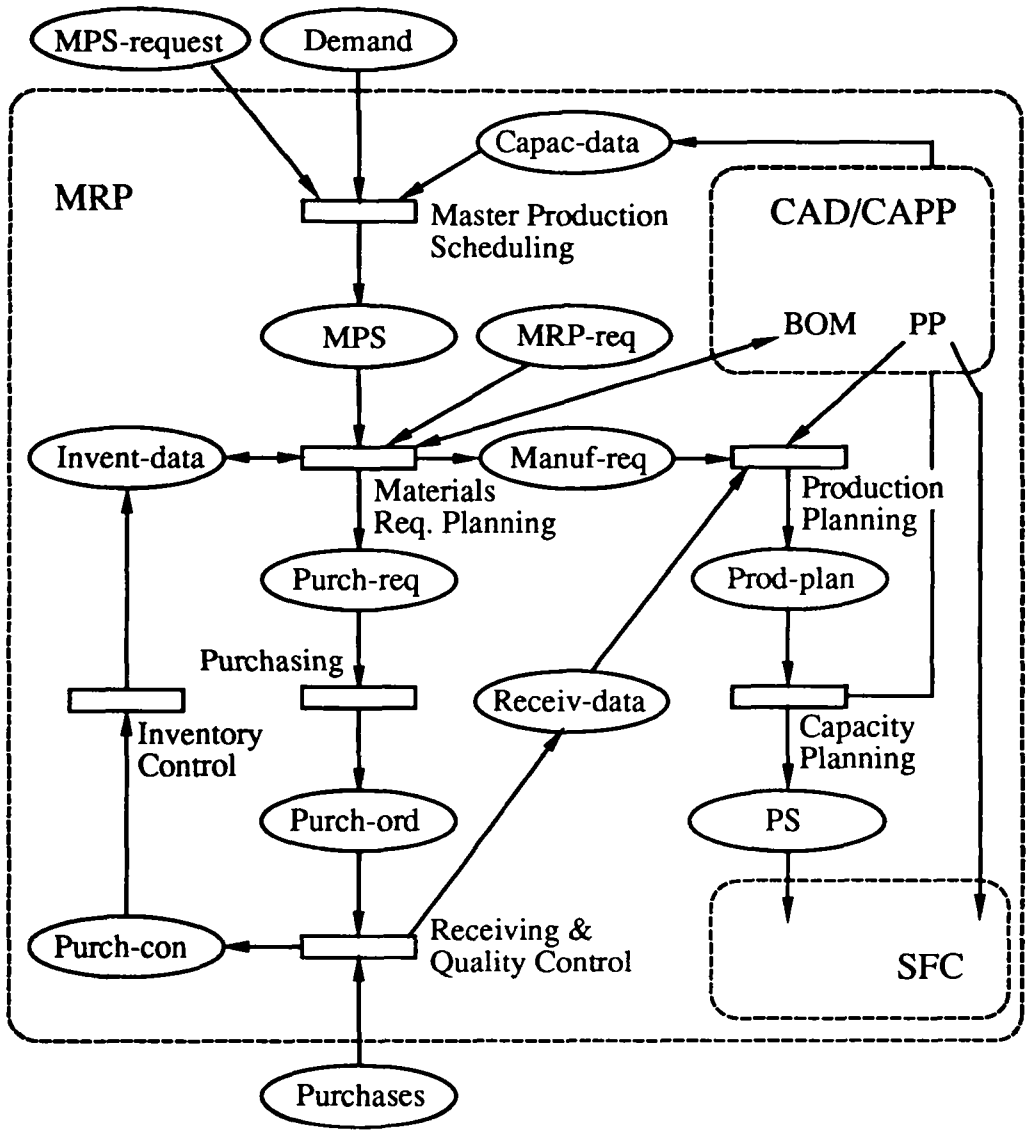
Fig. 5: Life Cycles of some CIM Objects

Bottom-up modelling involves a behaviour composition step in which parts of the life cycles of different components have to be combined to describe the overall behaviour of the process which utilize them. This step can then be seen as a reverse engineering step, which integrates existing models into more complex ones.

Although these steps have been described in a serial order, they must not be considered as strictly sequential, but can be executed partially in parallel and usually within several iterations for the sake of validation. In the M*-OBJECT approach, top-down steps (dealing with functional description refinement), outside-in steps (proceding both backward and forward), and bottom-up steps (modelling components and the related life cycles) are then integrated in an iterative strategy in which *what* is done in the object organisation is clearly stated. It must be noted that the resulting organisation nets *should not contain procedural specifications* which describe *how* functions are performed. Procedural aspects are taken into account in the conceptual design phase of the methodology to produce the executable specification of the environment.

The diagnosis method is a step-by-step method for guiding interviews with the users. The method indicates the kind of questions users should be asked to evaluate the actual operations of functions commonly found in production systems (such as production planning, inventory control, scheduling, process control, shop floor control, ...) [22]. Interviews are usually taped for subsequent reference and analysis, and diagnosis results are reported in a written document.

The bottom-up analysis is performed by means of interviews involving department managers but also supervisors and end-users. People are asked to describe processes in which they are involved as well as components (e.g. documents, forms and files) they use. Usually, the step requires at least three meetings per enterprise function: one to make the interview and the function diagnosis with the director or supervisor (1 to 2 hours), one to model the function with users (2 hours or more), and a short one to discuss the results obtained with function personnel (1 hour). However, since this step can be highly iterative, more meetings (or longer meetings) may be required depending on the complexity of the function to be analysed. Figure 6 illustrates the first level of specification for the ABC company in which only the MRP II function has been detailed into its major processes.



MPS-request : Master Production Scheduling request
MPS : Master Production Schedule
MRP-req : Materials Requirement s Planning request
Manuf-req : Manufacturing requirements
Purch-req : Purchasing requirements
Purch-ord : Purchase requirements

Capac-data : Capacity data
Invent-data : Inventory data
Receiv-data : Receiving data
Prod-plan : Production plan
Purch-con : Purchase control data

Fig. 6: First Level Specification of the ABC Company

Each function in the net of Fig. 6 can be further broken down. For instance, Fig. 7 provides the organisation net for the Materials Requirements Planning (MRP) process. Each execution of the net will result in a given set of orders to acquire materials or components (Purchasing Requirements), and a given set of manufacturing orders to make parts or products (Manufacturing Requirements). Each of the activities (Plan Gross Requirements and Plan Net Requirements) can be further decomposed, if required.

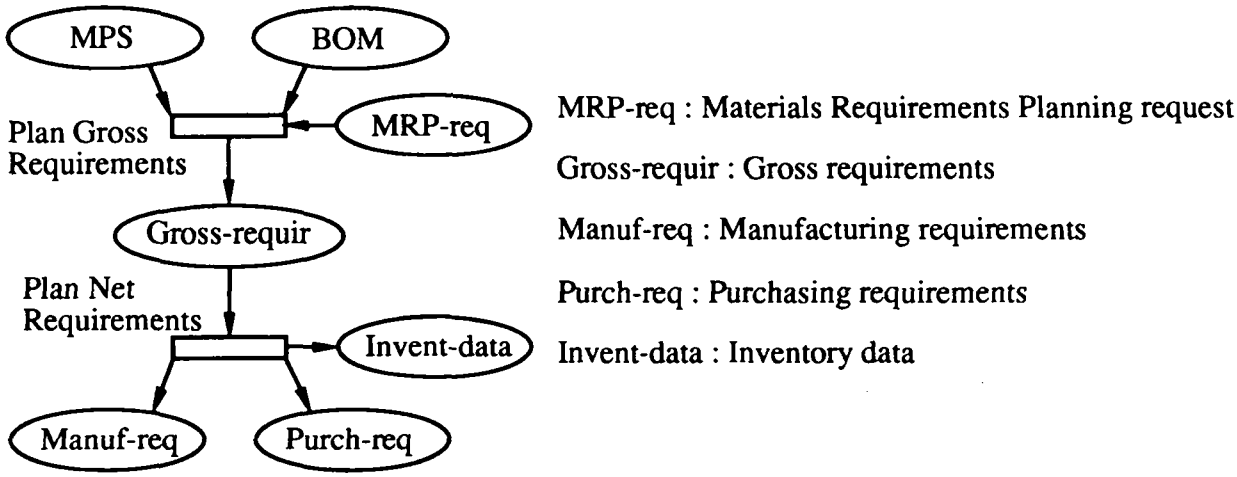


Fig. 7: Material Requirements Planning Process

Figure 8 shows the refined net for the Shop-Floor Control (SFC) environment. The Production Schedule coming from the Capacity Planning function of the MRP II environment is first decomposed into a detailed schedule to take into account availability of machining and handling machines at the shop-floor level and then released for execution (and executed by the flexible manufacturing cell). Routings indicate the actual operations to be executed by machines for a given part. Shop-Floor status (SF-status) indicates the status of each machine and the operations which have been executed (of course, the execute process can also be further decomposed).

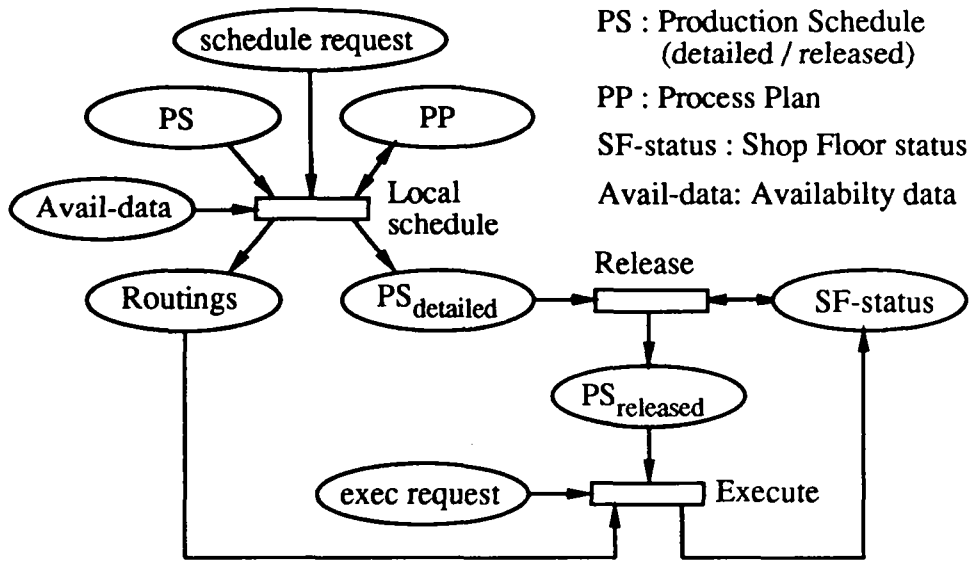


Fig. 8: The Shop Floor Control Environment

The overall architecture of the AS-IS Analysis task is shown in Fig. 9. The output of the AS-IS Analysis task consists of:

- a set of hierarchically correlated organisation nets and a set of life cycle nets which describe the functional structure of the object enterprise. These sets are globally referred as the enterprise AS-IS organisation schema;
- a diagnosis report which contains the results of the diagnosis steps and which will be used to identify opportunities for automation or enterprise improvement.

Top-down Analysis

1. Meet enterprise directors (top and middle management, decision-makers)
2. Consult the organisation chart
3. Identify all environments, their functions and related relationships (object flows)
4. Produce the general organisation net
5. Plan bottom-up analysis meetings

Bottom-up Analysis and Diagnosis

6. Analyse enterprise functions starting from production/service functions
   foreach function
   - 6.1 interview supervisors and users
   - 6.2 make the function diagnostic
   - 6.3 identify functions/processes/activities inputs, outputs, and agents
   - 6.4 build the life cycle nets of relevant components
   - 6.5 build the organisation nets which describe the organisation behaviour
7. Write the AS-IS Analysis report

Fig. 9: AS-IS Analysis Steps

7. The TO-BE Analysis Task

In the TO-BE Analysis task the overall structure and operations of the enterprise must be evaluated and eventually restructured. On one hand, the effort of analysts will be directed towards the identification of anomalies or malfunctions in the operations of the object organisation. On the other hand, managers and engineers will try to identify opportunities for automation or modernisation of the most critical environments. Then, proposals will be made to restructure the existing organisation into a new, more productive, or more efficient organisation. One proposal will have to be selected and approved by management. Finally, the new model of the organisation will have to be designed.
The M*-OBJECT methodology deals only with automation projects involving a database in their implementation. Therefore, the TO-BE Analysis task as described in this document is biased from an information system viewpoint.
The TO-BE Analysis task involves two steps:

1. *Anomalies Analysis*: This step is based on results of the AS-IS Analysis diagnosis report. Its aim is to produce a list of problems or anomalies in the functions of the organisation and to help in the formulation of an amendment plan. The models and the diagnosis elaborated during the AS-IS Analysis can be used to identify:
   (a) structural anomalies (e.g. wrong definition of a function);
   (b) operational anomalies (e.g. non-productive functions);
   (c) information-based anomalies (e.g. inadequate forms for data input, useless data flow).
   A report must be written in order to specify the anomalies and how they can be corrected (amendment plan). The report is then submitted to managers who will come back with an approved plan for restructuring part or all of the existing organisation, i.e. the list of which functions will be modified and/or automated and which will not.
2. *TO-BE Organisation Specification*: On the basis of both the AS-IS organisation schema and the approved plan for restructuration, the description of the altered functions in the new organisation must be produced. Moreover, a migration plan from the AS-IS to the TO-BE organisation must be established to define the sequence of actions to be taken.

It now becomes obvious that the AS-IS Analysis must be as accurate as possible, and must reflect the opinion of more than one person to provide a reliable support for timely management decision-making. Moreover, the decision will be based on a global understanding of the complete operations of the company. This decision will be followed by the specifications of the new system which will be expressed in terms of the same organisation model than the one used for the AS-IS Analysis. Then, the

specifications will be passed to engineers for implementation. This is the reason why the methodology makes use of specification models and methods that have been designed to be useful to and easily understood by managers, engineers, and technical people.
The output of the TO-BE Analysis task consists of a report which contains the TO-BE organisation schema, i.e. a set of organisation nets and life cycle nets which describe the functional structure of the new organisation, and the migration plan, i.e. the list of enterprise functions which will be automated with, eventually, the specification of the intermediate applications that will exist during the migration from the actual organisation to the future organisation supported by the information system.
The overall architecture of the TO-BE Analysis task is summarised in Fig. 10.

Anomalies Analysis

1. Identify anomalies
   foreach function
   1.1 identify structural anomalies
   1.2 identify operational anomalies
   1.3 identify information processing anomalies
2. Propose an amendment plan and submit it to management
3. Elaborate the restructuration plan

TO-BE Organisation Specification

4. Restructure functions
   foreach function to be restructured: define the new organisation nets and life cycle nets
5. Specify the migration plan

Fig. 10: TO-BE Analysis Steps

So far, we have discussed the methodology in the case of enterprise improvement situations. Obviously, the methodology can also be used for the design of new systems from scratch, although this is a seldom case. Since no AS-IS Analysis results are available for this kind of situation, the specifications of the new organisation will be based on management decisions and previous experience of engineers.

8. The Environment Specification Task

The aim of the Environment Specification task is to provide:

1. a framework, i.e. a set of guidelines, constraints and objectives, within which the information system implementation will take place;
2. enough information to begin the information system development.

Thus, the Environment Specification task is a logical link between the Organisation Analysis and the Conceptual Design phases. Its first step is the Design Team Set Up, within which a database committee is first formed. The design team has to perform the rest of the task, but more importantly, it has to perform the Conceptual Design and the Implementation Design phases of the information system under development, to choose the physical database management system (DBMS), and to plan the installation of the physical system.
The design team must not be too large to remain efficient. Conversely, it must not be too small. Indeed, it must be very representative of all applications and/or users that will be concerned by the database system to ensure that their data requirements will be taken into account. Usually, it is made of five to eight persons. One is a computer specialist and an expert on information system analysis and database technology. This person will chair the design team. The other persons assist in the design of the information system. They do not need to be data analysis experts but they must understand database technology and database design methodologies (such as M*-OBJECT in particular). What is really important is that this group of people be very representative of all departments, applications, and user functions affected by the future information system.

The design team is not necessarily identical to the team which did the Organisation Analysis phase. In fact, in practice it is never the case since the enterprise analysis is better performed if the analysts come from outside the enterprise, while conceptual and implementation design must involve people working in the day-to-day operations of the enterprise. However, if any continuity can be ensured, this will strongly augment the chance for success of the project. This continuity can be assumed by the team leader who might have been involved in the organisation analysis. Of course, the continuity is better ensured if both analyses are performed by the same consultants of a service company.
Once the design team has been established, its first activity is the analysis of both the migration plan reported in the TO-BE Analysis report and the management directives. An Information System plan is then prepared [16]. Usually, it contains: (a) the information system boundaries, i.e. a list of subparts of the information system to be built and the description of the interfaces interrelating them, (b) the data architecture, describing the basic ideas on which future databases will be constructed, and (c) the priorities, i.e. the sequence of implementation steps of information systems and databases.
A Feasibility Study of the information system plan is then carried out. It is dependent on organisation, technology, and economic factors [16]. Organisation factors are concerned with the agreement between the organisation plan and personnel and managerial objectives. From a technical point of view, the organisation plan must be evaluated in order to determine which computer resources (central processing units, disk space, terminals, network equipment, software, ...) must be engaged to have the applications running smoothly. Economic factors are considered in the context of a cost-benefit analysis. The feasibility study produces the description of the *system environment* within which information system implementation will take place.
Since the M*-OBJECT methodology is information system design-oriented, we will not discuss the aspects related the feasibility study and the cost-benefit analysis. In the remainder of the paper, it is supposed that other "ad hoc" methodologies have been applied and a system environment description has been produced in which technical constraints and priorities are clearly stated. In our case, we simply suppose that a database management system (object-oriented or not) will be used to physically implement the information system under development.
In the subsequent step, Requirements Collection Planning, organisation environments are reconsidered from an application design point of view, and the activity of the design team consists in identifying and collecting (information-oriented) system requirements from which data structures and activities will be specified in detail. Two main classes of system requirements are considered:

1. Formatted requirements: They refer to structured descriptions that can be described by means of a limited number of syntactic rules, and can be classified as (although the following list is not at all exhaustive):
   - Forms for data acquisition and output description or display (e.g. customer order forms, delivery slips, computer screen layouts, reports, etc.);
   - Record formats from data files (e.g. payroll file, bill of materials files, customer files, etc.);
   - Data Division from COBOL programs or other similar languages;
   - Data schemata defined with data description languages of pre-existing databases.

   Usually, the focus of formatted requirements is placed on data structures and their properties.
2. Non formatted requirements: They result from interviews and written descriptions. Such requirements are usually expressed by means of natural language sentences. Usually, the focus of non formatted requirements is on procedural description of enterprise operations (procedures, policies, document flows, etc.). They can be used to precisely describe, along with organisation nets and life cycle nets, the enterprise functions (processes and activities) to be automated.

First of all, the design team prepares a collection plan. For each environment, the collection plan identifies the requirements to be collected, as well as the users from whom detailed data and processing requirements about the functions that must be automated can be obtained. A complete organisation chart is essential for this step because not only process supervisors but also foremen, workers and end-users must be involved. The selection must be done in such a way that for each process (which is a component of a function to be automated) a significant set of interviewees is provided.
Requirements Collection is the final step of the Environment Specification task.
Figure 11 gives an example of a formatted document (in this case, a process plan form as used in the production planning process).

Version n#:..... Designer name:..... Date:...../...../.....

Part-id : FICBLOC
Part-name: blade

Drawing# : 125
Process-plan# : 27

sequence#	op-code	Operation name	Machine code	NC program code
1 121		turn diameter	1440-lathe	FICBLOC_L1
2 214		drill hole	W50-machining centre	FICBLOC_M1
3 127		measuring	8202-B&S	FICBLOC_CMM1

Fig. 11: The Process Plan Form

Figure 12 illustrates the data division of a COBOL program which processes a BOM (bill of materials).

```
IDENTIFICATION DIVISION.
.....
DATA DIVISION.
  01  ITEM.
    02  PARTNO       PIC 9(6).
    02  DESC         PIC X(20).
    02  LEAD_TIME    PIC 9(3).
  01  COMPRISES.
    02  LEVEL        PIC 9(2).
    02  ITEM         PIC X(10).
    02  SUBITEM      PIC X(10).
    02  QTY          PIC 9(2).
PROCEDURE DIVISION.
.....
```

Fig. 12: A COBOL Program for BOM Processing

Figure 13 is a recording of an interview discussing the activities of the Execute process of the Shop Floor Control environment of the ABC company.

The overall architecture of the Environment Specification task is summarised by Fig. 14.

Environment: Shop Floor Control Process : Execute User : .....

A fabrication lot is a batch of similar parts to be produced. Each occurrence of a part of the same lot is a workpiece made of some material which is manufactured according to a predefined sequence of machining operations specified by a process plan.
A workpiece must be mounted on the appropriate device to be processed on a given machine with a tool of a certain type. After each operation, the workpiece is prepared for the following operation until the process plan is completed.
.....

Fig. 13: The Execute Process Description

Design Team Set Up

1. Select team members
2. Analyse the migration plan and management directives
3. Prepare the Information System plan

Feasibility Study

4. Analyse the information system plan from organisation, technology, and economics points of view
5. Produce the system environment specification

Requirements Collection Planning

6. Prepare the collection plan
   foreach environment
   - 6.1 establish the list of functions and interviewees
   - 6.2 establish the list of forms and file specifications to be collected
   - 6.3 plan requirements collection

Requirements Collection

7. Collect requirements
   foreach environment
   foreach function
   - 7.1 interview users and collect relevant information
   - 7.2 collect forms and file specifications used in the function activities

Fig. 14: Environment Specification Steps

9. Conclusions

In this report, the M*-OBJECT methodology for information system analysis and design of CIM environments has been presented and its Organisation Analysis phase discussed.
The meta-model of M*-OBJECT, based on the concepts of agent, event, function (process and activity) and component, has first been introduced. It is the glue which ties together the various phases of the methodology, and it provides a language close to the language used by application users.
Concerning the Organisation Analysis phase, its aim is to provide system analysts with precise enough tools and methods to model the current status of the organisation (or part of it), to establish deficiencies in the organisation structure, and to define a new structure to correct deficiencies. Then, precise system requirements can be defined which will serve as a basis for the next phase of the methodology, i.e. the Conceptual Design.
An Organisation Model has then been provided to describe the organisation of an object enterprise. It is based on the concepts of object views and functions. It is used to specify the flow of control, the flow of objects (materials or data), and the organisation structure of the part of the enterprise being analysed.
Organisation Analysis is a crucial phase in the design of an information system of a CIM environment since the information system must conform to the objectives and constraints imposed by the CIM system.Thus, precise requirements on the information system must be derived from the analysis and specification of the CIM environment being developed. This is the aim of the Organisation Analysis phase of M*-OBJECT as discussed in this report.

10. References

[1] Bravoco R.R., Yadav S.B. "Requirement Definition Architecture - An Overview" *Computers in Industry*, **6**: 237-251, 1985.

[2] Bravoco R.R., Yadav S.B. "A Methodology to Model the Functional Structure of an Organization" *Computers in Industry*, **6**: 345-361, 1985.

[3] ESPRIT Consortium AMICE (Ed.) *Open System Architecture for CIM*, Springer, Berlin 1993.

[4] Williams T.J. "A Reference Model for Computer Integrated Manufacturing Systems from the Viewpoint of Industrial Automation" Proc. 11th IFAC World Congress, Tallinn, 1990.

[5] Rolland C. et al. "Information System Development: A Survey" Res. Report MASI 91.38, Lab. MASI - Université Paris 6, 1991.

[6] Ross D.T., Schoman U.E. "Structured Analysis for Requirements Definition" *IEEE Trans. on Software Engineering*, **SE-3**(1), 1977.

[7] Teichroew D., Hersey E. "PSL/PSA: A Computer Aided Technique for Structuring Documentation and Analysis of Information Processing Systems" *IEEE Trans. on Software Engineering*, **SE-3**(13), 1977.

[8] De Marco T. *Structured Analysis and System Specification*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[9] Rolland C., Richard C. "The REMORA Methodology for Information Systems Design and Management" in: Olle T.W. et al. (Eds.) *Information Systems Design Methodologies: A Comparative Review*, North-Holland, Amsterdam, 1982.

[10] Mylopulos J., Yang D., Fry J.P. "A Language Facility for Designing Interactive Database-Intensive Applications"*ACM TODS* **5**(2), 1980.

[11] TEMPORA, ESPRIT II Project, Concepts Manual, 1989.

[12] DiLeva A., Vernadat F., Bizier D. "Information System Analysis and Conceptual Database Design in Production Environments with M*" *Computers in Industry*, **9**: 183-217, 1987.

[13] DiLeva A., Vernadat F., Bizier D. "Enterprise Analysis and Database Design with M*: A Case Study" *Computers in Industry*, **11**: 31-52, 1988.

[14] Chen P.P. "The Entity Relationship model: Toward a Unified View of Data" *ACM Trans. on Database Systems*, **1**(1): 9-36, 1976.

[15] Berio G., Di Leva A., Giolito P., Vernadat F. "The M*-OBJECT Methodology for Information System Design in CIM Environments: The Conceptual Design Phase" Res. Rep. INRIA, 1993.

[16] A. Blokdijk, Blokdijk P. *Planning and Design of Information Systems*, Academic Press, 1987.

[17] Gane C., Sarson T. *Structured Systems Analysis: Tools and Techniques*, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[18] Parnas D.L., "On the Criteria to be Used in Decomposing Systems into Modules" *Communications of the ACM* **15**(2), 1972.

[19] Paulson D., Wand Y. "An Automated Approach for Information System Decomposition" *IEEE Trans. on Software Engineering* **18**(3): 174-189, 1992.

[20] Yourdon E., Costantine L.L. *Structured Design*, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[21] Richter G., Durchholz R. "IML-inscribed High-level Petri Nets" in: Olle T.R. et al. (Eds.) *Information System Design Methodologies: A Comparative Review (CRIS 1 - IFIP WG8.1)*, North-Holland, Amsterdam, 1982.

[22] Boyer L., Poirée M., Salin E. *Précis d'Organisation et de Gestion de la Production* (Chap.VIII), Les Editions d'Organisation, Paris, 1982.

Unité de Recherche INRIA Lorraine
Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Antenne de Metz
Technopôle de Metz 2000 - Cescom - 4, rue Marconi - 57070 METZ (France)

Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Rocquencourt Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399