PPCsm^{2}e – Production Planning and Control System for Small and Medium Manufacturing Enterprises

Gabriel Neagu, Constanta Manoliu and Octavian Constantinescu

National Institute for Research and Development in Informatics
8-10 Averescu Avenue,
71316 Bucharest
ROMANIA
E-mail: gneagu@ici.ro

Abstract: There is a large consensus in considering small and medium sized enterprises (SMEs) as being the industrial entities best adapted to current trends and requirements in the global economy. Therefore one of the major challenges of today information technology and communication field is to support continuous enforcement of SMEs competitive potential. The COPERNICUS '94 project RaPOrt (#1191, led by Archimedia, Ltd., Greece) aimed at providing a set of tools and a methodology to support the implementation of the production planning and control (PPC) software for industrial SMEs with particular emphasis on the needs imposed by Central and East European SME structural characteristics.

The paper presents the *PPCsm^{2}e* software for manufacturing SMEs , developed at ICI Bucharest within the RapORt project. The Generic Prototyping Approach has been adopted as a methodological framework for the system development, considering its compliance with current requirements in the complex system engineering. The balanced implementation of the generic modelling and prototyping paradigms, as well as the support for gradual system development based on a coherent integration framework are of special interest from the RaPOrt objectives point of view. After a GPA overview the paper details the modelling constructs at the generic system level for the *PPCsm^{2}e* software: (1) the application domain reference model, a PMS architecture like solution adapted to SMEs PPC specificity , and (2) the generic system object oriented conceptual model. The second part of the paper presents the *PPCsm^{2}e* functionality: system administration, technical and commercial data administration and PPC related functions (i.e. master production scheduling, material requirements planning, capacity requirements scheduling and order progress reporting). The current state of *PPCsm^{2}e* software implementation and the objectives for its further development are tackled in the final Section of the paper.

1. Introduction

SMEs have a sound specificity from the point of view of production planning and control requirements. Small firms with loose organizational structure need by far less complicated computer-aided production management systems [14]. Therefore, it would be more profitable to simplify the organizational structure, as well as the production management software systems based on object oriented data modelling techniques and aggregation of decision models rather than start re-engineer the factory. Also, such companies are not in the position of making pressure on their suppliers to get them deliver on a Just-in-Time basis.

The Information Technology fast advances represent new challenges to innovation at the enterprise level. According to [8] major factors generating difficulties for SMEs in gaining benefits from the area of information technology are the following:

- incompatibility of newly introduced systems with the existing computer infrastructure;
- rapid changes the SMEs are forced to react to, because of the ever shorter product life-cycles, while IT projects might have a duration that is longer than expected;
- lacking of IT professionals within the enterprise;
- limited resources available for starting innovative IT projects.

In Romania, like in any other of Central and East European countries, SMEs are considered to be the main support for industrial re-structuring of the former big, state-owned enterprises. Their organization and development represent a priority objective in the transition to market economy. Nowadays, major hindrances of such enterprises activity would include: unsteady market demands for their products, relatively plain structure of products, reduced capital and working staff. If compared with the SMEs in Western countries, differences will markedly reside in the available equipment and technologies, level of automation, the invested capital. The use of information technologies aims at narrowing this gap, providing software tools and systems to increase the productivity, to enforce the control of resource utilisation, to reduce costs, and to improve reaction to market demands through production planning, scheduling and control.

The awareness of Theory - Practice gap in the area of production planning and scheduling systems in small manufacturing companies (see [6] for the UK market), as well as the necessity for keeping the cost of such systems as down as SMEs can afford, made the Copernicus 1994

RaPOrt project #1191 be launched . *RaPOrt - Rapid Prototyping of Object-oriented Production Planning and Control Systems for Industrial SMEs* aimed at providing a low-cost, customizable support for production planning and control solutions in manufacturing SMEs. The project consortium associated Archimedia Ltd. from Greece - as project coordinator, ICI Bucharest, Institute "Joseph Stefan" from Ljublijana and the UTAL S.A. pilot enterprise in Bucharest. This paper presents the ICI Bucharest contribution to this project, namely the *PPCsm²e* software. Firstly proposed in [9], the Generic Prototyping Approach-GPA, has been adopted as a methodological framework for the *PPCsm²e* development. The GPA belongs, as other original prototyping and generic modelling approaches do, to the class of modelling approaches, sharing the preoccupation with abstract description of the future system as a determinant feature of the development process. The second Chapter of the paper is devoted to GPA description , with an emphasis put on system development processes and integration support for system components instantiation according to this approach. The next two Chapters discuss the main GPA generic modelling constructs of *PPCsm²e* software: the reference model of the application domain and the conceptual model of the future generic system. Chapter 5 provides an overview of the *PPCsm²e* software, with focus on its main functionality. Some conclusions are drawn and further work is briefly defined in the last Chapter.

2. The Project Methodological Framework

2.1 GPA Orientation

The Generic Prototyping Approach complies to some major orientations in complex system engineering : involvement of end-user in the system development process starting from the early requirement definition phase; open system architecture as a prerequisite for a gradual improvement of the system functionality and performance; object-oriented analysis and design with major benefits as to the natural matching between the reality and the model, the abstraction power, the structuring flexibility and the interaction flexibility, the robustness of object-oriented design with respect to later changes in system requirements.

The *prototype* is a work product rather than a final one, with limited functionality as compared with the target system (TS). It facilitates answering questions about the target system based on the end-user's involvement as member of the development team, who is responsible for the final result. On the other hand, the generic model facilitates the passing over of the responsibility for the TS development to the end-user [3]. The degree of involvement in this process depends on the solution opted for in the generic model implementation, which ranges from a generic system (GS) to an environment for interactive model building. GPA is a combined approach built around the *generic prototype (GP)* concept. GP is a prototype-like preliminary version of the future GS, with a limited (but workable) functionality. At the same time, the GP genericity consists in its deliverable character as a final product, which incorporates the host system (HS) capabilities for TS generation.

2.2 GPA Processes

The GPA life-cycle includes two processes: the prototyping and integration processes, dealing with *GP development*, and the instantiation process for *GP utilization* and *maintenance*. They are represented in Figure 1, as coordinates of the conceptual framework provided by the instantiation and derivation processes of CIM-OSA [15].

The **prototyping process** has a diagonal orientation as reported to this framework, with an emphasis put on the OO approach for the requirements definition (RD) and design specification (DS) phases. The *RD phase* consists of two steps. The first step is devoted to the analysis of the application domain (AD) which the future GS belongs to, providing the AD reference model (AD/RM). During the second step of this phase the *GS conceptual model (GS/CM)* is developed. Chapter 2.3 provides more details about these GPA generic modelling constructs. The *DS phase* delivers the GS architecture (GS/A) and the GP thorough design specifications (GP/DS). During the third, *implementation (I) phase*, *particular prototypes (PPs)* are developed as GP building blocks, in order to validate implementation solutions for top GP/DS modules. Structural and cognitive resources are implemented at the partial level of the CIM-OSA cube, considering their capabilities for integrating PPs.

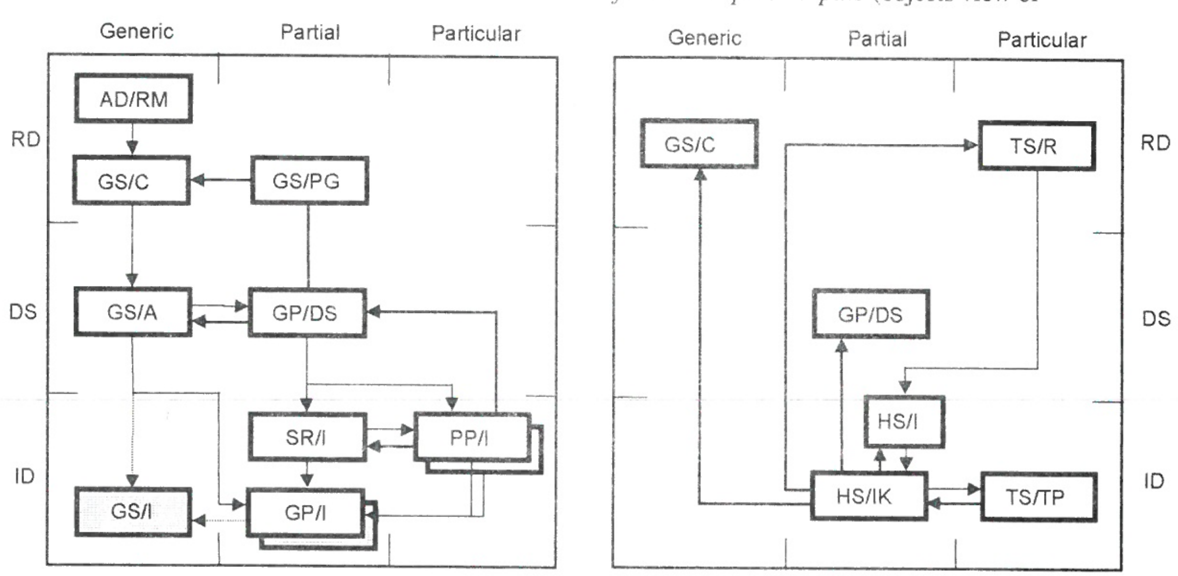
The **integration process** aims at implementing the GP based on a relevant set of PPs. Further, the GS implementation phase may start, using a set of GPs developed according to the same GS/CM and the same GS/A. It is recommended that a careful analysis of this phase convenience, based on cost/effectiveness criteria is done, which explains its optional mode in Figure 1.

The **instantiation process** deals with the generation of concrete target systems (TSs) using GP as a HS. The process is driven by the instantiation parameters (HS/IP) which express the TS requirements (TS/R). The result of the instantiation process is the target prototype (TP). As usual, while under implementation, it is subject to further development, in order to meet specific TS requirements, which are not covered by the GP. Therefore TS is the result of two consecutive processes: the GP instantiation and the TP implementation. On evaluating the TS in relation with the initial TS requirements, this sequence may be reactivated in an iterative way as a TS tuning mechanism. Such feedback links between phases and processes are represented with fat arrows in Figure 1.

2.3 GPA Generic Modelling

According to [16] in the CIM environment, RM is meant to define requirements common to all implementations, but it is independent of the specific requirements of any particular implementation. It has an important role in maintaining the coherence during the analysis phase and in evaluating the completeness and extensibility of a concrete system.

In the GPA environment the AD/RM model building is oriented on basic constructs of the enterprise modelling [2]. The *domains* defines the part of the enterprise which is relevant to achieving a defined set of business objectives. A domain is defined by a set of *domain processes* (which fulfil domain objectives under some domain constraints), the *classes of objects* they handle and the *relationships* with other domains (boundary definition). Domain processes are the starting point of function decomposition using business process and enterprise activity constructs. The *business process (BP)* defines what has to be done in a defined logical control sequence in order to obtain the desired result. The *enterprise activity* defines the functionality of a domain of the enterprise and is described in terms of *function inputs/outputs* (objects view of



a) GP development: prototyping and instantiation processes

b) GP utilization and maintenance: instantiation process

AD - Aplication Domain
GS - Generic System
GP - Generic Prototype
PP - Particular Prototype

SR - Structural Resources
HS - Host System
TS - Target System
TP - Target Prototype

RM - Reference Model
CM - Conceptual Model
SA - System Architecture
GPS - GPs Set

DS - Detailed Specification
I - Implementation
R - Requirements
IP - Instantiation Parameters
IK - installation Kit

Figure 1. GPA Processes

the enterprise objects which are operated on during the enterprise activity), *control inputs* (objects or information elements which control and/or constrain the execution of the enterprise activity), *control outputs* (a set of information elements which define the ending status of the activity), *resource inputs/outputs* (objects used/objects returned by the activity during its execution) and *transformation function* (tasks describing the functionality of the activity).

The building process starts with developing AD business architecture (AD/BA), using *domain* and *business process* concepts. The AD/BA construct is further refined through functional decomposition operating with the CIM-OSA *activity* concept, using the *IDEF 0* method [1].

At the GPA generic level the second major modelling construct is GS/CM, which includes the object, control, and user-interface sub-models. The first two models are similar to the corresponding system analysis sub-models of the OMT method [13]. The third one reveals the importance of user-interface for the system architecture design, according with the general principle stating that a user interface should be, in terms of both system architecture and development, treated apart from the application itself [11]. This specificity resides in the particular consistence and relevance of the interface to the negotiations between user and developer. Recently the configuration of the OO system modelling components has been similarly viewed. For example the *HP FUSION* OO modelling system includes such tools as the *object model*, the *object interaction graph* and the *interface model* [5]. The functional analysis and decomposition included in the AD/RM construct provide, within the GPA framework, the support for defining menus and options. Further the information about classes of objects, their attributes and relations specific to the CM/O sub-model is used for a draft specification of different forms, messages and reports. Obviously, the emphasis is put on the structural view of the interface, using the object model formalism. Its dynamic view usually complies with the standard solutions specific to the technological environment selected for the future system development and implementation phases. Depending on the environment facilities, the development of an operational version for the user interface will be recommended.

2.4 GPA Integration Framework

The GPA integration process provides solutions to the GP incremental development through the integration process of a relevant set of PPs. In the context of complex system engineering, the integration of system components generated during the GPA instantiation process has to take place. The GPA integration framework (GPA-IS) aims at providing a coherent formalism to express both integration requirements and integration solution in GPA specific terms [10].

For the requirements definition, the **connectivity dimension** of GPA-IS operates with connectivity levels. The degree of connectivity for two target components (TCs) is defined based on the vicinity relation between the functional areas of their host GPs. The following levels are considered:

- direct-connectivity level (DcL) referring the target components instantiated from GPs belonging to the same GS/CM;
- tight-connection level (TcL) characterizing target components with distinct host GPs, which are included in the same AD/RM;
- loose-connection level (LcL) relevance in case of target components being generated from GPs which represent adjacent AD/RMs.

The **compatibility dimension** of the GPA integration framework defines levels of solution to meet integration requirements and is based on the GP specific architecture levels:

- resource level (RL) relevance to CTs sharing resources; here counts the information resources, while procedural and cognitive resources are only used depending on functional similarities between given CTs;
- service level (SL) referring the components sharing services, i.e. solutions do not differ in case of similar functionalities;
- interface level (IL) defining the compatibility between different CTs according to the user interface specific elements, and dealing with both semantic (user options) and syntactic (interface structure and implementation solutions) aspects.

Based on this two-dimensional framework, different integration strategies have been articulated. Their role is to illustrate possible integration schemes for system components, balancing the integration requirements with integration solutions in the context of a complex system development. Obviously, not all system components are to be generated as GP target components. These strategies primarily address the TCs, where the expected compatibility levels may be automatically programmed via a GPs instantiation process, provided that the corresponding GPs implement these levels. Nevertheless, given their meaningful terminology, the integration strategies are recommended for all system components, irrespective of how they have been developed. For non CTs components the implementation of the required compatibility levels should be realised along with their development. Table 1 presents the set of integration strategies defined according to the GPA integration framework and covering all connectivity levels.

Table 1. GPA Integration Strategies

Compatibility levels:	Connectivity levels: DcL	TcL	LcL
IL	Required	required	required
SL	Required	prioritary	optional
RL	Prioritary	optional	irrelevant

In case of DcL, the involved components are to be found at the same hierarchical level in the overall system architecture. This means, they serve the same user group, behave similarly, and are implemented on the same platform or on others alike , sharing a significant amount of resources.

For the TcL connectivity, the target components actually belong to the same subsystem, devoted to the given AD. They are working on the same platform, which normally requires high compatibility at the service architecture layer. As the user community is still homogeneous, the visibility level is also important. The amount of commonly used resources is smaller than that in the previous case, which makes the RL compatibility optional, depending on the importance and dynamic features of the involved resources.

The LcL connectivity addresses the case when the components belong to distinct subsystems. Consequently, they are usually implemented on different platforms, even in different working environments. The commonly used resources are at a minimal level, if at all. An open system environment is powerful enough to solve the connectivity problem via the required resources transfer among components. This makes the RL compatibility irrelevant. Further the SL compatibility is considered as optional and mainly concerns resource management aspects. The decision of its implementation depends on the compatibility of the technological support. Despite the potential diversity of profiles and requirements which such components' end-user communities may present, the TL compatibility is thought to be a minimal requirement for their integration, so that a common "look and feel" attitude at the system level should exist.

3. PPC/SME Reference Model

Based on the SMEs specificity and in order to ensure a higher integration degree for its components, an aggregated application domain has been adopted for the *PPCsm²e* software. This AD centres on the production planning and control process, but also includes functionally adjacent domain processes as presented in Figure 2.

For the PPC domain, as a reference solution, the PMS (Production Management System) architecture has been adopted by the project consortium. PMS had been developed under the IMPACS (Integrated Manufacturing Planning and Control System) project (ESPRIT - 2338) and further used as a reference in COSIMA (Control Systems for Integrated Manufacturing) project (ESPRIT - 477) [7]. If reported to the system architecture taxonomy proposed in [8], PMS is a business architecture which helps clarify the mission and nature of the given domain. Being aware of the specific functional and organisational features of the production activity in SMEs, some remarks should be made on the PMS architecture:

- Business planning is primarily addressed for its sales and financial support; limited manufacturing resources of SMEs do not allow to properly operate with a manufacturing strategy; instead some qualitative objectives of the manufacturing activity may be formulated , as guiding criteria for lower levels of production management.
- Master production schedule seems not to be so vital as in case of bigger enterprises because of higher manageability of the production environment. Its importance is mainly tributary to the production type, while evolving towards a make-to-order one-of-a-kind production.

- At the shop floor level an easy-to-overview environment facilitates the so-called "eye management" and relieves of dispatching and monitoring activities . Also, the scheduling function focusses on an optimum capacity loading given the limited technological alternatives and the lack of flexibility in job ordering at this level.
- The assumption that adjacent data are provided by other specialized components of the enterprise information management system does not hold in case of SMEs with their low level of IT implementation and few

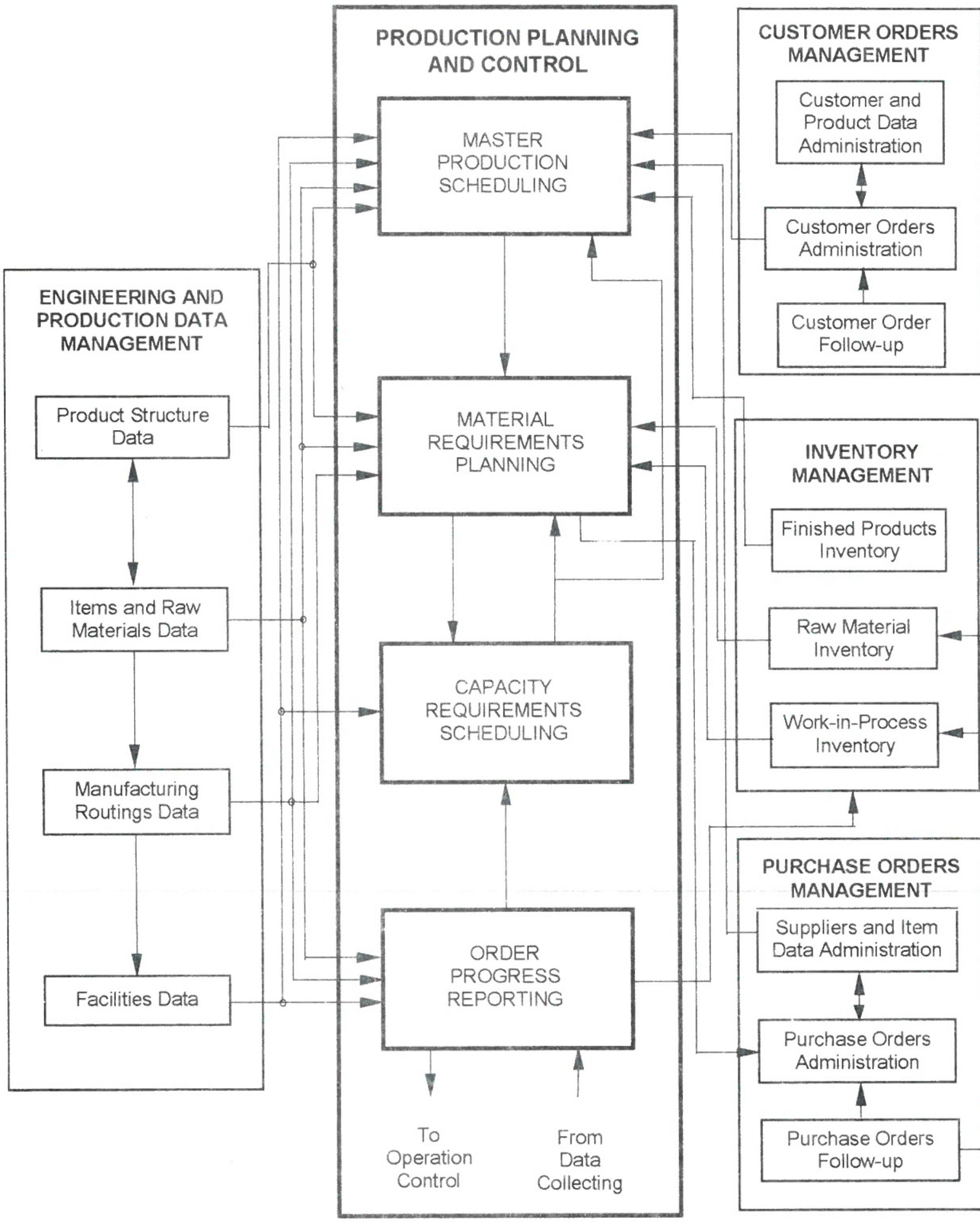


Figure 2. PPC/SME Business Architecture

possibilities for integrating IT modules.

According to GPA Requirements Definition phase, the AD/RM construct takes two phases: business architecture definition and functional decomposition of the domain processes.

The PPC/SME business architecture shown in Figure 2 has been proposed to comply with the specificity of the application domain. First, it is a simplified, aggregated model as compared to the initial PMS architecture, based on the four PPC stages: master production scheduling, material requirements planning, capacity requirements scheduling and order progress reporting. Secondly, the proposed architecture includes, in order to provide functional autonomy to the future system as to input data availability, some adjacent application domains (engineering, production, inventory, customer and purchase order data management).

The conclusions of the application domain analysis are given in the following, in terms of objectives identified for each particular domain process, as well as classes of objects relevant to the envisaged software functionality. In Figure 2 business processes for each domain process are shown. For adjacent domain processes only the business processes supporting the functional autonomy of the future system are considered.

Production Planning and Control

Objectives: to ensure the carrying out of contracted obligations in accordance with the stipulated due dates and quality requirements based on planning and control of efficient utilization of production resources.

Classes of objects: manufacturing requirements, performance criteria, planning horizons, priority criteria, manufacturing planning variants, batches, released orders, schedules, dispatching rules, shop floor current state, abnormal situations.

Engineering and Production Data Administration

Objectives: to supply coherent engineering and production data.

Classes of objects: products, component items and raw-materials, engineering drawings, technological operations, manufacturing routings, production capacities (machines, machine groups, work centers , workcells), tools.

Purchasing

Objectives: to obtain the required quantity of materials by the date specified, at the lowest possible price, consistent with the required qualitative level.

Classes of objects: suppliers, offered raw - materials and items, purchase requirements, purchase order.

Inventory Management

Objectives: to elaborate a policy for optimal level inventory ensuring: (1) low investment in finished products, components, work-in-progress, raw - materials, maintenance parts, tools; and (2) non-interrupted production flow.

Classes of objects: finished products inventory, raw-materials inventory, work-in-progress, inventory input/output transactions backlog.

Customer Order Servicing

Objectives: to handle customer orders and provide up-to-date input data for production control.

Classes of objects: customers, supplied finished products, customer orders.

For the second development step of the PPC/SME RM, the first level activities in the *IDEF 0* decomposition hierarchy are presented:

AD1-MPS:

A1.1 Product requirements evaluation

A1.2 Master schedule generation

A1.3 Master schedule validation

AD2-MRP:

A2.1 Dependent item gross requirements determination

A2.2 Dependent item net requirements determination

A2.3 Item lead time determination

A2.4 Material requirements determination (at the time bucket level)

A2.5 Lot sizing

A2.6 Purchasing and work orders generation

AD3-CRP:

A3.1 Available capacity identification

A3.2 Capacity requirements definition

A3.3 MRP feasibility validation

A3.4 Capacity requirements scheduling

A3.5 Work graphics generation

AD4-OPR:

A4.1 Aggregated follow-up data presentation

A4.2 Abnormal situations identification

A4.3 Production activity reporting

Based on the results of the Reference Model development, the functional configurations that may be defined for the *PPCsm²e* software are those presented in Table 2.

Table 2. *PPCsm²e* Functional Configurations

Ordered product input data source	co	co	co	co	of	of
Master Production Scheduling	y	y	n	n	y	y
Material Requirements Planning	y	y	y	y	y	y
Capacity Requirements Scheduling	y	n	y	y	y	n
Order Process Reporting	op	it	it	op	op	it
Variant no.	1	2	3	4	5	6

where:

co – customer orders
of – orders and forecasts
y/n – activate or not the given function
op – operation level
it – item level

4. PPCsm²e Conceptual Model

According to the Generic Prototyping Approach, the second step in the requirements definition phase is devoted to the object-oriented RD/RM construct development, including object (structural), dynamic (behavioural) and user interface sub-models (see Chapter 2.3). The *System Architect* CASE tool [12], which implements the OMT method, supported such development.

4.1 Object Sub-model

Given the complexity of the object sub-model, the following sections have been defined and the corresponding OMT object diagrams have been drawn:

- **TECHNO** for production area entities:
  - item, with its specializations: component, final, as well as raw-material, standardized, detail, assembly;
  - item structure;
  - item routing;
  - technological operation: productive (processing), move, setup, inspection;
  - item route operation;
  - department;
  - workshop: job shop, flow shop, parallel machines.
- **COMM** for commercial area entities:
  - partner: supplier, customer;
  - order: purchase, delivery (requested final item);
  - item: purchased, customer.
- **MPS** for function entities specific to *Master Production Schedule*:
  - item requirements: requested final item, forecast final item;
  - summarized item requirements;
  - requested internal component item;
  - work in progress inventory item;
  - requested purchased component item;
  - purchased inventory item;
  - required work center capacity;
  - work center;
  - MPS record;
  - master production schedule.
- **MRP & CRS** for entities describing Material Requirement Planning and Capacity Requirement Scheduling functions:
  - MPS record;
  - requested purchased component item;
  - purchased inventory item;
  - purchase order;
  - internal component item;
  - requested internal component item;
  - work in progress inventory item;
  - shop order;
  - required work center capacity;
  - work center;
  - work center order;
- **INV** for inventory related entities:
  - inventory item: purchased, work in progress, inventory item, finished inventory item;
  - inventory transaction.

For each object diagram an Entity-Relationship diagram has been generated by means of SA tool. For each entity there are specified, among other attributes, the source class as well as private and foreign keys. Also different types of relationships (links and their cardinalities, inheritances, part of links) are represented. A special entity – like representation is used for many-to-many links. The ER diagram for the TECHNO section is presented in Figure 3.

4.2 Dynamic Sub-model

The dynamic sub-model is represented by the proposed OMT event trace diagrams. This kind of a diagram describes the behaviour of a group of related entities (classes of objects) according to a scenario (sequence of events). The meanwhile between events represents the state of the given object, while the transition between states represents the response to an event. The diagram shows each object as a vertical line and each event as a horizontal arrow from the sender object to the receiver object. Time increases from top to bottom, but it is only a sequence of events that is shown, not their exact timing. A set of event trace diagrams for the allowable scenarios that might occur, helps identify possible states for the involved objects and set up state diagrams at the object level.

The dynamic sub-model for the *PPCsm²e* system includes **MPS** (see Figure 4) and **MRP&CRS** event trace diagrams. For each diagram the System Architect provides a list of all the elements in.

4.3 User Interface Sub-model

Two types of constructs are included in this sub-model: menus and graphic screens. The **menu hierarchy** is detailed in an SA report indicating the name and type (submenu or menu item) of each node in this graph, as well as the name of the graphic screen corresponding to each terminal node. For **graphic screen diagrams** the name and type of each included symbol are provided in a dedicated SA report. Figure 5 illustrates both components of the user interface. Given the importance of this sub-model at the GS/CM construct level, the SA support has been of special interest in the designing, versioning and administration of all related information.

5. PPCsm²e Overview

5.1 Application Area

The *PPCsm²e* software has been developed for small and medium manufacturing enterprises with order-based or order- and forecast-based production. For those who should make decision on production control, this software product is a useful tool as it permits the analysis of the conditions under which the production tasks will be successfully completed.

At various planning levels (global, detailed, operation oriented), the availability of resources (raw-materials, production capacities) is checked and actions are encouraged in order to obtain a feasible production program:

- product oriented action: allocate a batch to another period of time, with more capacities available and within a slack time which is not beyond the due date;
- decision-maker action: either modify the priority of customer orders or change the quantities brought in by forecasts processing; in either case, restart simulation as many times as necessary to reach a satisfactory program;
- management action: take regulatory measures (organize extra shifts, subcontract overwork) in order to fulfil the contractual provisions.

5.2 Characteristics

Generic

The product design has been based on a reference model for discrete production enterprises. Man-machine dialogue let the customer's requirements be considered and the proper version of the product be delivered.

Versatile

Given the aforementioned planning levels, a modular structure has been adopted, to meet various requirements. During the set-up procedure functional variants are available for end user's selection.

A non-sophisticated solution

- the aggregated production management model generates less sophisticated data requirements as to volume and structural complexity;
- given its design, the product is an easy-to-operate one and does not suppose expert knowledge:
  - there is an embedded coding system which the end user should not be aware of (only descriptions are provided by the user interface for selection and validation purposes),
  - all operations related to an item can be executed on the same form,
  - help information is provided at both general and menu item levels.

Low cost investment

- minimal hardware and software requirements,
- non-expensive adaptation and operation.

5.3 Functional Configuration

The software product has been developed under the Access 97 DBMS and includes the following three major components: system administration, technical and commercial data administration, and production planning and control.

5.3.1 System Administration

This component executes the following functions:

- define user groups and individual users
- initialise tables
- save and restore data
- display log files

For each function an interface to the corresponding management function of Microsoft Access 97 is provided, as a support for novice users of Access.

5.3.2 Technical and Commercial Data Administration

Production planning procedures are enforced by means of a significant database. This component enables such usual operations as: add, modify, delete, navigate, display, print. The database comprises two main categories of data:

- technical data:
  - organization of the production areas in an enterprise (work center configuration),
  - items (finished products, assemblies, details, raw -materials, purchased and standardized materials),
  - structure of finished products and assemblies,
  - manufacturing technologies,
  - technological operations,
  - timetables,
  - planned machine stops.
- commercial data:
  - partners (customers and suppliers),
  - orders/contracts (for purchase and delivery),
  - inventory control (finished products, raw- materials),
  - input/output transactions (raw-materials, finished products).

5.3.3 Production Planning and Control

Checking on Inputs

Manufacturing demands may originate from two sources:

- customer orders/contracts,
- orders and forecasting.

Forecasting is to estimate demands well in advance, referring the orders' due dates and the so far delivered quantities of each finished product at a previous time. Taking forecasting as a single working mode is not customary with small and medium sized enterprises, which this software product is meant for.

On placing one customer order before another, the following priority criteria will bear: customer importance, customer credibility, order value, ordered quantity.

Horizon Definition and Master Production Scheduling

The period to be covered by the program is defined. Manufacturing demands belonging to this period are selected from the list provided at the previous step.

Checkings are on:

- fitting an order fulfilment to a period,
- covering the period with purchased items,
- covering the period with processing capacities.

Following the second checking, information is obtained about:

- items for which there exist raw - materials in store,
- items for which raw -materials have run out, but they were purchased during the last 12 months (including name of supplier, quantity and the latest transaction date),
- items for which raw- materials have not been purchased before.

Following the third checking, information is obtained about:

- manufacturing demands which are likely to be met,
- products incorporating items with non-available processing capacities (also orders including these products),
- items with non -available processing capacities,
- overloaded work centers (extra capacity required, indicated in minutes).

Given the results, and for reaching a proper solution, the decision-maker may either take regulatory measures or resume the programming algorithm, while changing some input data regarding:

- order priorities,
- predicted quantities.

Material Requirements Planning

A production program for pre-established short periods of time is generated. It considers the duration of the manufacturing process, the average time necessary for executing operations and the production volume (medium, small, single) over the period defined at master production scheduling. This function takes several stages:

- shop orders settling: shop orders result from either master production scheduling or orders/orders and forecasting (if master production scheduling has been excepted).
- production program development: in order to distribute the quantity of purchased and internal items, the Critical Path Method is used.
  A graph is generated for each finished product, with nodes as events, and arcs as activities. The events reveal the product structure by levels and items. Activity duration means to put together the technological times of all operations on an item. As a result of the critical path generation, the earliest and the latest moments for operation starting are defined, as well as the slack time for each item. The quantities allocated to the same period (usually the period stands for a week) are summed up.
- submission of results: reports are provided concerning time distributed gross and net requirements for both internal and purchased items.

Capacity Requirements Scheduling

Material requirements planning as done at the previous stage has ignored the capacity limits. This time the capacity demands are well defined for operation at the work center level loading. This function is optional: a user is free to give up this module if he/she finds the operational program as irrelevant.

Operation assignment per work center is defined according to:

- the yields of material requirements planning (the earliest and the latest start dates, slack time);
- the timetable of operations to be performed at a work center and planned machine stops.

The principles and rules which apply are the following:

- backward like operation assignment, starting from the delivery day on;
- no overlapping of operations (a new operation will only start when the preceding one ends);
- in case of a work center overload, the operation is deferred up to the moment of an available capacity turning up, but within the slack time imposed by the manufacturing demands;
- if no available capacity turns up, or the slack time is zero, order items are overlooked.

This function output consists in:

- work center orders,
- work center loading diagram,
- out of capacity work centers (required capacity flow in minutes)

Order Progress Reporting

This function addresses the following levels:

- *operation*: a manufacturing plan at operation level has been drawn up; reporting is on its starting and ending moments, as well as on the quantity produced;
- *item*: effective no matter the level of material requirements planning (per item or operation); updating intervenes on the starting and completion times and on the item quantity specified in the shop order;
- *finished product*: is done by the "Transactions" component in the Commercial Database. The quantity is updated in the Inventory Control module and can be assigned to a customer order and deducted from the inventory level after booking the dispatch note.

The module submits reports on:

- current state of orders progress,
- items with later due dates,
- required orders from a given moment up to the end of the period.

Some screen samples of the *PPCsm*^{2}*e* software are presented in Figure 6.

5.4 Implementation Requirements

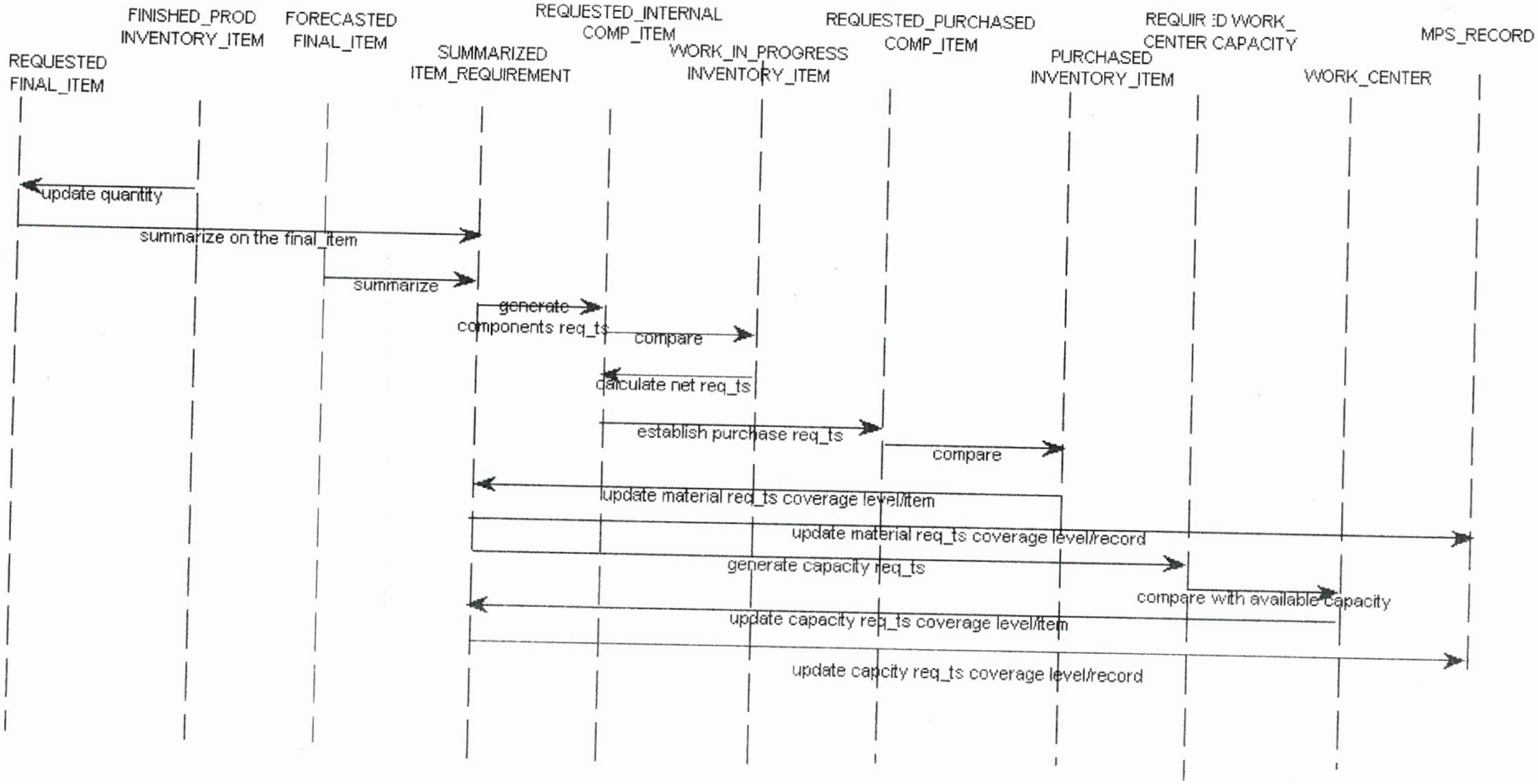
Hardware: PC Pentium computer with 16Mb RAM, 50 MB disc space, 3.5" FDD, printer; **Basic Software**: Windows '95 or Windows NT operating system.

6. Conclusions

PPCsm²e is a generic, versatile and low-cost solution for the PPC implementation in Small and Medium Manufacturing Enterprises, developed according to the Generic Prototyping Approach. Its genericity is based on the GPS system modelling constructs, encapsulating the major system design decisions. The business architecture of the addressed application domain is a PMS like solution adapted to the SMEs specificity: an aggregated application domain including adjacent to PPC data, which provides sub-domains and a simplified hierarchy of PPC functions. The conceptual model of the generic PPC/SME system is object oriented and covers its static, dynamic and user interface views. Being implemented under the *System Architecture* CASE tool, the GS/CM construct plays the role of a working exploratory prototype, according to the classical taxonomy proposed in [4]. The versatility of the system is supported by its role of generic prototype, as starting point for the instantiation process, delivering target prototypes adapted to different PPC functional configurations. Their low-cost characteristic is of special interest for this class of enterprises. *PPCsm²e* software has been developed under Microsoft Access, for small to medium ranged PC configurations. There are two pilot implementations of this software: at UTAL S.A. and PIFATI S.A., two enterprises located in Bucharest. Further development of the software aims at enlarging the covered functional area of the enterprise, evolving towards an integrated enterprise-wide management system, as an ideal solution for SMEs .

REFERENCES

1. BRAVOCO, R.R. and YADAV, S.B., **A Methodology To Model the Functional Structures of An Organization**, COMPUTERS IN INDUSTRY, 6, 1985, pp. 345-361.
2. CEN TC 310 WG1, **An Evaluation of CIM Modelling Constructs**, COMPUTERS IN INDUSTRY, 24 (2-3), 1994, pp. 159-236.
3. FILIP, F.G. and NEAGU, G., **CIM in Continuous and Discrete Manufacturing: Object-oriented Generic Modelling**, INT. J. CONTROL ENGINEERING PRACTICE, 1, 1993, pp. 815-825.
4. FLOYD, C., **A Systemic Look to Prototyping**, in R. Budde (Ed.) Approaches to Prototyping, SPRINGER-VERLAG, Berlin, 1984, pp. 1-18.
5. GRABOT, B. and HUGUET, P. **Reference Models for An Object Oriented Design of Production Activity Control Systems**, in L.M. Camarinha-Matos and H. Afsarmanesh (Eds.) Balanced Automation Systems. Architectures and Design Methods, CHAPMAN & HALL, 1995, pp. 311-318.
6. HALSALL, D.N., MUHLEMANN, A. P. and PRICE, D.H.R., **A Review of Production Planning and Scheduling in Smaller Manufacturing Companies in the UK**, PRODUCTION PLANNING & CONTROL, 5(5), 1994, pp. 485-493.
7. HIGGINS, P., LYONS, G. and BROWNE, J., **An Architecture for Production Management Systems**, CIM-Europe Workshop on Open Systems Architectures and Communications, Aachen, 1989.
8. MEIJS, C., **Narrowing the Gap of SME for IT Innovation with Enterprise Architectures**, in L.M. Camarinha-Matos and H. Afsarmanesh (Eds.) Balanced Automation Systems. Architectures and Design Methods, CHAPMAN & HALL, 1995, pp. 37-44.
9. NEAGU, G., **Generic Modelling Vs. Prototyping: An Object-oriented Approach to the Decision Support At the Shop Floor Level**, in R.S. Sodhi (Ed.) Advances in Manufacturing Systems: Design, Modelling and Analysis, MANUFACTURING RESEARCH AND TECHNOLOGY, 22, Elsevier, 1994, pp.19-26.
10. NEAGU, G., **Integration Levels in Generic Prototyping Approach**, Preprints of IFAC Conference on Integrated Systems Engineering, PERGAMON, 1994, pp. 477-482.



MPS- Event Trace diagram

Figure 4. Dynamic Sub-model Representation

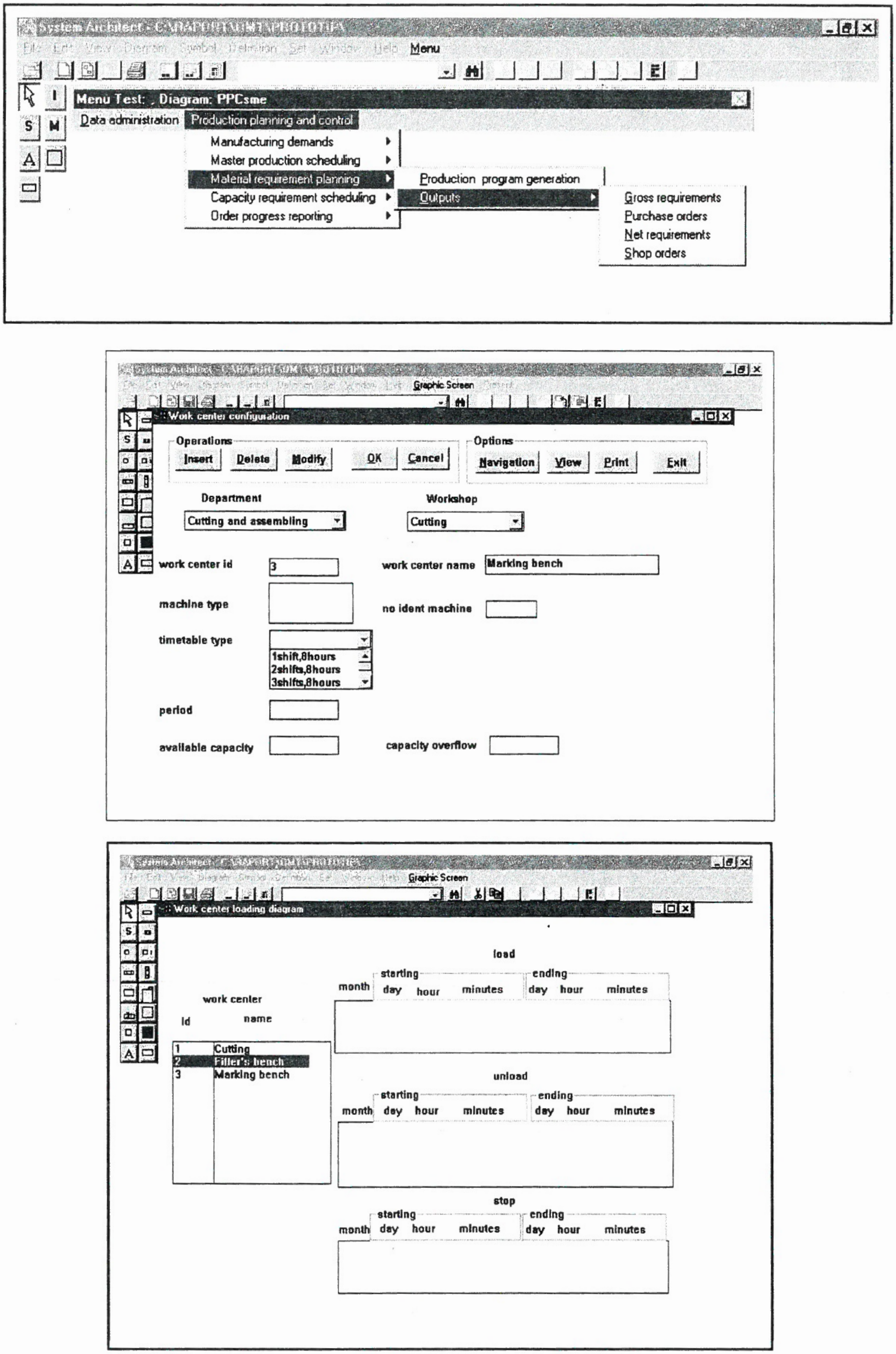


Figure 5. User-interface Sub-model Representation

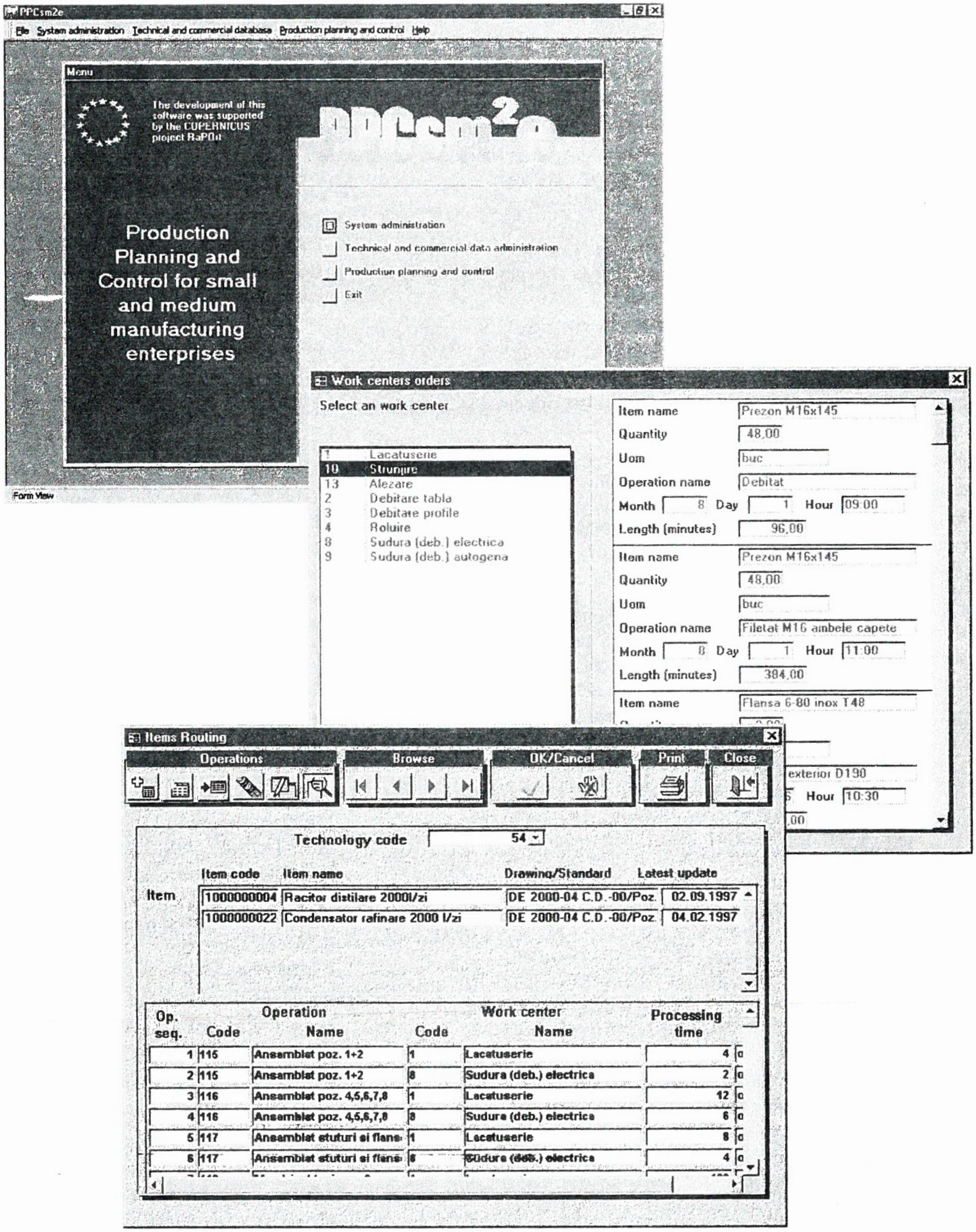


Figure 6. PPCsm^{2}e Screen Samples

PMS Architecture

The proposed reference architecture is structured on 3 levels and 5 stages of production management. Issues involved at each PMS level are the following:

- strategic level (the planning horizon from one to five years): determination of the product to be manufactured, the matching of products to markets and customers' expectation, design of the manufacturing system;
- tactical level (one month to one year): generation of detailed plans to meet the demand imposed by the master production schedule, which is realistic in terms of capacity and materials available;
- operational level (real-time to one week): taking the planned orders from a MRP system and managing the manufacturing system in quasi real-time to meet these requirements.

Business Planning, Requirements Planning and Production Activity Control stages belong to the strategic, tactical and operational levels, respectively. The Master Production Scheduling stage is placed at the border between strategic and tactical level, while the Factory Co-ordination stage is a specific PMS stage at the border between tactical and operational architecture levels. The content of each production management stage is detailed in the sequel.

- **Business Planning**
  From the point of view of manufacturing aspects, the aim is to develop the manufacturing strategy for supporting the overall business goals of the enterprise.
- **Master Production Scheduling (MPS)**
  - is a statement of the anticipated manufacturing planning in terms of *what, how many* and *when* to produce ;
  - involves demand management, rough-cut resource planning and final assembly scheduling;
  - is based on end items, specific customer orders or some group of end items and product options, depending on *make-to-stock, make-to-order* or *assemble-to-order* production environment;
  - *the planning horizon* is the cumulative lead time including engineering design time (in case of a make- to- order company), material procurement time, production lead time and assembly lead time;
  - the scheduled demand is derived from past demands (in case of a *make-to-stock* environment), from the backlog of customer orders (*make-to-order*) or from a mixture of forecasting and customer orders (*assembly-to-order*);
  - its *verification* ensures that the master schedule aggregates up to the production plan; its *validation* consists in checking that the available materials and capacity resources are adequate to meet the requirements;
  - the frequency of its *maintenance and change management* is determined by the forecasting cycle (usually monthly).
- **Requirements Planning**
  - it may involve Material Requirements Planning (MRP) and Manufacturing Resource Planning (MRP II) techniques;
  - MRP deals with material planning, inventory control and BOM control functions; its aim is to convert the MPS for end products to a detailed schedule for raw- materials and components used in the products;
  - MRP II is an extension of MRP, with an emphasis on capacity requirements planning.
- **Factory Co-ordination**
  - is a set of procedures regarding the planning of the flow of products at plant level, in order to reduce the complexity of control task at lower levels;
  - includes four main modules:
  - plant level scheduler for the co-ordination of the flow of batches between workcells through production allocation (including batch sizing for each product), bottlenecks management (at the workcell or individual resource level) and plant level schedule generation;

- plant level dispatcher for the implementation of the plant level schedule; a special concern of this module is for handling late batches;
- plant level monitor for providing relevant data on real-time status of the shop floor (the amount of daily finished batches, throughput times, and work in progress levels) to the plant level dispatcher and monitor.

- **Production Activity Control**
  - operates at the workcell level, in a time horizon of between one month and real-time;
  - includes the following functions:
    - short term scheduling (one day to one week) as a basis for order releasing;
    - dispatching, as the final determination of job sequencing at work center level;
    - monitoring, acting as a reporting mechanism;
    - materials transport function for shop floor transportation devices control;
    - process control function, addressing specific production equipment such as CNC machines and robots.