29

Reference Models for an Object Oriented Design of Production Activity Control Systems

B. Grabot, P. Huguet°**
**Laboratoire Génie de Production, ENIT,*
Avenue d'Azereix-BP 1629, 65016 Tarbes Cedex FRANCE
Phone: (33) 62 44 27 21 Fax: (33) 62 44 27 08
°AEROSPATIALE, Centre de Recherche Louis Blériot
12, rue Pasteur-BP 76, 92152 Suresnes Cedex FRANCE
Phone: (33) 1 46 97 35 22 Fax: (33) 1 46 97 32 59

Abstract
The reference models of high level analysis and design methods for manufacturing systems, such as CIM-OSA or GRAI, provide a good support at the conceptual and organisational levels, but do not help in to integrate existing software, or to develop and reuse code. Object Oriented methods are good candidate for the improvement or integration of the analysis, specification and development phases of manufacturing systems: they are recognised as providing an efficient support in order to shorten the code development phases, and to develop modular and reusable software. Nevertheless, most of the applications developped in the past years focus on very low level control structure design, and there is a lack in supporting tools for complex production activity control architectures. We suggest in this paper to define modular reference models of PAC modules in order to provide a framework for adaptable PAC systems design. The prototype of support system which development is now in progress intends to be a real "workstation" for the PAC systems designers, in order to better take into account important factors such as reuse of models and code, and integration of existing software.

Key words
Production Activity Control, reference models, Object Oriented Analysis

1 INTRODUCTION

The Production Activity Control (PAC) system of a manufacturing system has to manage the short term planning, execution and monitoring activities required to process manufacturing orders on the physical resources of the workshop. Manufacturing systems have to react more and more quickly, efficiently and at low cost to changes in the production context, either internal (e.g. machine failure) or external (e.g. change in orders).

In order to be efficient, the PAC system must be perfectly adapted to the workshop it controls. This sets the problem of the integration of already existing software, and leads most of the time to the development of costly additional software. Reference models of manufacturing systems have been proposed for quite a long time in analysis or design methods,

L. Camarinha-Matos et al. (eds.), *Balanced Automation Systems*

in order to provide a framework that should facilitate and improve this design phase. The better known of these methods (CIM-OSA and GRAI-GIM), emphasizing the conceptual and organisational levels of design, are shortly described in section 3. On the other hand, Object Oriented methods have had a great success these last years for numerous reasons, one of them being that they allow both system modularity and code reuse. In order to improve their efficiency, we suggest in this paper reference models of PAC modules (operationaly organised as clusters of classes) that can be organised in order to design versatile control architectures. The implementation of these classes in a CASE tool supporting the HP FUSION method is in progress, in order to provide a support for the system designer: the suggested design methodology of the PAC system is detailed in section 4.3.

2 THE PRODUCTION ACTIVITY CONTROL CONTEXT

Production activity control is now usually defined as the set of activities permitting short-term production and inventory control of a shopfloor as well as adapting production to the various disturbances that may occur within the shopfloor or its environment.

A lot of research works has been performed on different types of production activity control systems (see for instance (Jones and Saleh, 90), (Lyons et al., 90), (Van der Pluym, 90) or (Hynynen, 88)), most of them oriented towards the definition of a framework in order to develop a particular solution. Most of the small or medium enterprises can not afford the development of complex specific software for their PAC systems, but even industrial groups like AEROSPATIALE prefer to integrate existing software in their PAC systems, since very good tools are now available on the marketplace. Since no complete solution is available, the PAC system has then to be built using existing software (e.g. schedulers, follow-up software, controllers) integrated with specific developments. A study has then been launched in AEROSPATIALE in order to provide a tool that could facilitate the integration, maintenance and evolution of the PAC system and support the specification and design of PAC systems.

In order to satisfy these requirements, we need reference PAC models that can be easily particularised for a given application, i.e.:

- models that cover the whole abstraction cycle of the PAC system design, i.e. from conceptual aspects to code development,
- modular models (in order to facilitate integration),
- reusable models, in order to reduce development time and cost for different PAC designs.

In order to define these models, we have first studied the existing analysis and design methods for manufacturing systems.

3 METHODS AND MODELS FOR MANUFACTURING SYSTEM DESIGN

3.1 The CIM-OSA methodology

The first goal of the CIM-OSA project (AMICE, 89) is to provide an architecture of the manufacturing system that helps the integration of computer applications. This reference architecture has been widely spread through the well-known "CIM-OSA cube" that models the various aspects of the life cycle of a CIM system along:

- a "derivation axis", describing life cycle steps: requirements (definition), design (specification) and implementation (description),
- a "generation axis", composed of various views of the system: function, information, resource and organisation,
- an "instantiation axis", describing levels of implementation: generic, partial and particular.

The methodology uses a top-down approach in order to describe the three modelling levels of the derivation axis with the different views of the generation axis, passing then through the various instantiation levels. Modelling tools adapted to each step are suggested, e.g. a hierarchical approach using building blocks for the requirement definition model (which have some analogies with SADT), or the entity/relationship model for the specification of the information view.

A major interest of CIM-OSA is to provide a comprehensive framework for the life cycle of a manufacturing system. In comparison with our own particular objectives, this framework is nevertheless too high level to provide support for PAC system integration, since PAC has many specificities compare to the rest of the decisional sub-system of the manufacturing system.

3.2 The GRAI and GRAI-related methods

The original GRAI method (Doumeingts, 84) aims at the analysis, diagnosis and improvement of production management systems. The method uses three different models:

- a conceptual model, that splits the manufacturing system into physical, decisional and information systems,
- a structural model (GRAI grid), that describes the decisional system in charge of production management along two axis: a functional axis (the columns) and a temporal axis (the lines) composed of the various decision making horizons. Each cell of the grid defines a decision centre, that can be precisely described through the GRAI-nets.

The final models result from a comparison between a top-down approach (the grid) and a bottom-up approach (using the GRAI-nets). Rules are then applied in order to find the dysfunctions of the system described.

The GRAI method was extended under the name of GIM (for Grai Integrated Method) (Roboam, 88). The modelling phases are roughly similar to those of the Merise Method, dedicated to the information system development (Tardieu et al., 83). Although the GRAI method is closer to the PAC system than CIM-OSA concepts, it is more dedicated to the highest levels of production management than to the operational levels of the manufacturing system.

Another extension of the GRAI method has been made to take into account PAC problems in manufacturing systems, called GRAICO (Fénié, 94). GRAICO uses an original method based on graph theory in order to analyse the flows in the physical systems (and not only in order to hierarchise the resources like in most of the other methods). The PAC-module structure suggested remains close to the GRAI grid, and consistency rules between physical and PAC systems are suggested.

In spite of their efficiency for the improvement of the production management system, methods such as GRAI, GIM or even GRAICO do not match our objectives perfectly, since they stand quite far from software development and do not sufficiently take into account the existing experience in PAC system design.

4 OBJECT ORIENTED DESIGN METHOD FOR PAC DESIGN

Object Orientation is now widely recognised as providing reusable models and code, and improving the process of software development. One of its interests is that it provides a natural implementation of the physical entities of the workshop using objects (Nof, 94), (Rogers, 92).

4.1 Use of OO methods for manufacturing system design

A lot of work has been performed on manufacturing systems modelling using object oriented methods. Most of these works describe classes based on the manufacturing entities, that can be instantiated for the design of control systems (see for instance (Smith and Joshi, 92), (Hinde et al., 92), (Rogers, 92), (Nof, 94), (Hopkins et al., 94)) but very few are interested in the control structure. Nevertheless, it seems clear that "flat" structures (i.e. structures with a single level of control) are not sufficient in all the cases (Rogers, 92). The problem of the control structure is clearly correlated to object oriented analysis in (Elia and Menga, 94), and in this work the chosen structure is the NIST model (Jones and Saleh, 90) based on successive aggregations of the physical resources through facility, shop, cell, workstation and equipment levels. This work is oriented on the problem of communication between modules, and not on the precise content of the control modules, this last point being very important in order to prepare software integration.

4.2 Reference models of a PAC module

Since one of our goals is to provide support for different types of workshops, we do not want neither to prescribe a particular control structure, nor to define a unique PAC module. Based on the works referenced in section 2, a PAC module may be defined through four main functions:

- to plan, that considers the production plan and the objectives transmitted by the upper decisional level, and details it,
- to react, that allows real-time adaptation of the decisions taken at different management levels each time it is necessary,
- to launch, that breaks down, synchronises and executes decisions taken within a PAC module,
- to follow-up, that collects all the results provided by the sensors or the lower decisional levels as well as the requests coming from outside the shopfloor, processes them according to their type and aggregation level and transmits them to other functions and other PAC modules.

Based on an analysis of representative workshops within AEROSPATIALE, we identified seven main types of modules which differ by the functions involved and their relationships (Huguet and Grabot, 94). For instance, a PAC module can be purely reactive at the lowest level (i.e. it has no planning facility) whereas a high level module may have no other reaction than performing a new planning: this is entirely compatible with the GRAICO approach, where high level decision are triggered by a time period whereas low level decisions can be triggered by events. A typical PAC module is shown in figure 1: function "plan" transmits a set of orders to execute to function "dispatch", which is in charge of resource requisition and synchronisation. If an unexpected event occur, the "react" function may modify the sequence of orders transmitted to "dispatch", trigger the "plan" function, or ask for reaction to an upper level module if the event sets drastically into question the general planning that comes from the upper level.

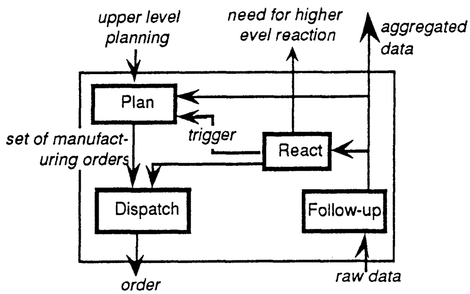

Figure 1 Example of PAC module.

These PAC modules permit to define an integration frame for existing software, but software development remains most of the time necessary, e.g. to support the "react" function which as to be closely adapted to the workshop characteristics. In order to facilitate this development and its reuse, these reference models have been implemented in an object oriented methodology: HP FUSION™.

4.3 Design methodology

Although most of the object oriented analysis methods are based on the same concepts, and use similar modelling tools, HP FUSION (Coleman et al., 94) has been chosen for the following reasons:

- it supports all the object orientation concepts (i.e. classification, polymorphism, inheritance...),
- it supports seamless development: same concepts are used from analysis to implementation, which allows iterative and incremental development,
- it covers broadly enough the different perspectives of the system development (e.g. static and dynamic aspects),
- efficient support tools can be found on the market (e.g. PARADIGM Plus™ and FUSION softcase™).

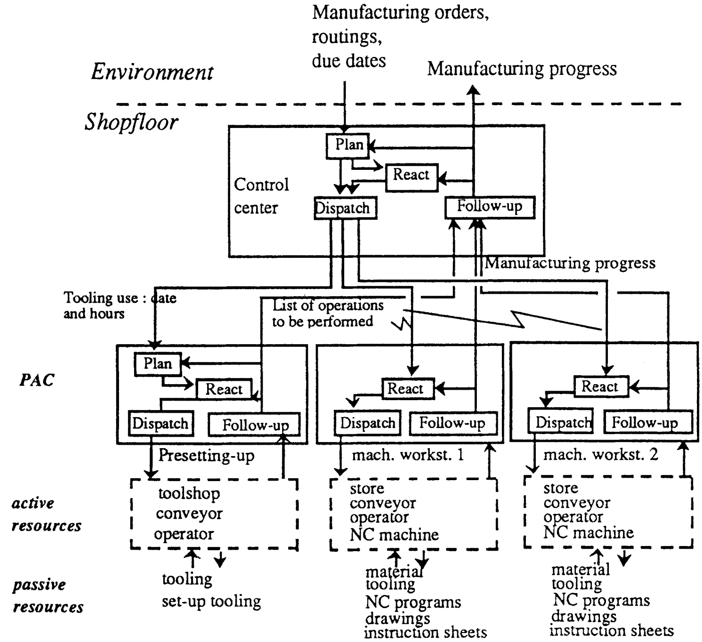

Figure 2 PAC model of an AEROSPATIALE workshop.

The main tools of FUSION are:

- the Interface model (IM). This model shows the main exchanges of information between the system and the agents acting in its environment.
- the Object Model (OM). This diagram adds the object formalism (instantiation, decomposition, inheritance) to the entity-relationship diagrams.

- the Object Interaction Graph (OIG). This graph shows the exchanges of messages between the objects described in the OM.

The PAC system design methodology defined at the moment follows the FUSION methodology:
- modelling of the relationships of the workshop with its environment, then the relationships of the PAC system with the physical resources through the Interface Model,
- modelling of the main resources of the workshop through instantiation of pre-defined classes in the Object Model,
- modelling of the PAC system through instantiation of classes deriving from the reference models in the Object Model,
- checking of the system behaviour through the Object Interaction Graph.

An analysis of the consistency between workshop and PAC models is performed through different sets of rules. Forty to fifty rules have been selected on the base of the first four models describing typical workshops of AEROSPATIALE. These rules are not definitive at the moment, and the knowledge base is subject to constant improvement on the base of further applications (Huguet and Grabot, 94).

4.4 Examples of models

Figure 2 shows the functional model of one of the studied workshops. A scheduling is first performed using a scheduler called SIPAPLUS (PRIOS). This schedule is then transmitted to the toolshop, where a planning of tool utilisation is performed on the base of the global schedule. Two machining workstations are controlled by purely reactive PAC modules, capable to manage small disturbances.

Example of reference models (or patterns) are shown in figures 3 to 5. These models provide a standard description of classical manufacturing operations, and can be modified for adaptation to specific cases if needed. An example of Interface Model is provided in figure 3, describing the passage from an operation to another on a machine.

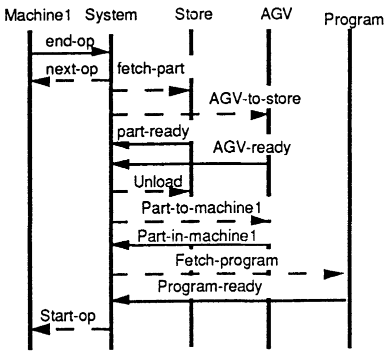

Figure 3 Example of Interface Model.

A partial object model is shown as an example in figure 4: this model shows the implementation a PAC module through objects: its components (planning system, dispatcher, follow-up system and reaction system) will then be split if the software that will support each of them has not already be chosen.

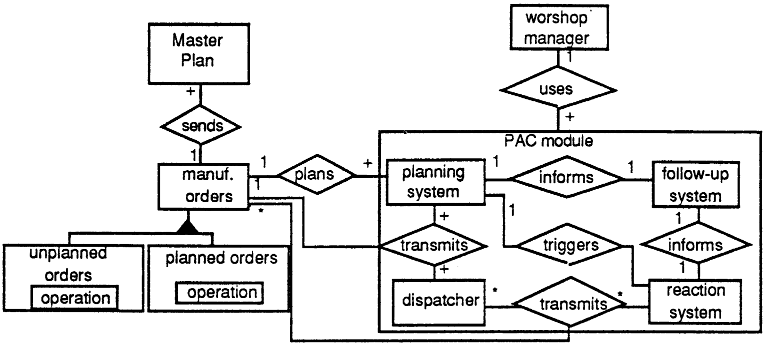

Figure 4 First level object model of a PAC module

Figure 5 shows an example of Object Interaction Model corresponding to the interface model of figure 3.

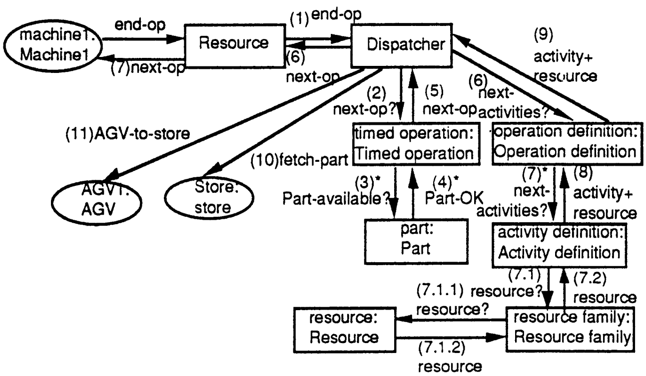

Figure 5 Object Interaction model

5 CONCLUSION

The use of Object Oriented analysis, design and programming methods is one of the elements that should help in order to design more easily and more quickly more efficient and better adapted PAC systems. This goal can only be achieved through reuse, which require reference models in order to provide a framework for the integration of existing software and specific development.

The first experiments performed with the reference models of the PAC modules that we suggest have shown that these models can be applied to a great variety of workshops. Furthermore, the implementation of these models through libraries of pre-defined objects facilitates and shortens greatly the design phase. Major improvements of the models and of the knowledge base that checks the workshop/PAC consistency are expected after other implementations, but it is clear that the support tool which development is in progress must be considered as a way to capitalise expertise, rather than as an automated way to design PAC systems.

6 REFERENCES

AMICE Consortium (1989) *Open System Architecture for CIM*, ESPRIT Project 688, volume 1, Springer Verlag, Berlin.

Codd, E.F. (1970) A Relational Model of Data for Large Shared Data Banks, *Communications of the ACM*, 13, **6**, 377-387.

Coleman, D. Arnold, P. Bodoff, S. Dollin, C. Gildchrist, F. Hayes, F. Jeremaes, P. (1994) *Object-Oriented Development - The Fusion Method*, Prentice Hall, Englewood Cliffs.

Doumeingts, G. (1984) Méthode GRAI : Méthode de conception des systèmes en productique, State Thesis, University of Bordeaux I.

Elia, G. Menga, G. (1994) Object-oriented design of flexible manufacturing systems, in Computer Control of Flexible Manufacturing Systems, Joshi S.B. and Smith J.S. Eds, Chapman & Hall, 315-342.

Fénié, P. (1994) GRAICO : méthode de modélisation et de conception des systèmes d'exploitation de systèmes de production, PhD thesis, University of Bordeaux I.

Fox, M. (1981) An Organizational View of Distributed Systems, IEEE Transactions on Systems, Man and Cybernetics, vol. SMC 11, **1**.

Hinde, C. West, A. Williams, D. (1992) The use of object orientation for the design and implementation of manufacturing process control systems, International Conference on Object-Oriented Manufacturing Systems, Calgary, Canada, 3-6 may.

Hopkins, J.M. King, R.E. Culbreth, C.T. (1994) An object-oriented control architecture for flexible manufacturing cells, in Computer Control of Flexible Manufacturing Systems, Joshi S.B. and Smith J.S. Eds, Chapman & Hall, 427-466.

Huguet, P. Grabot, B. (1994) Production Activity Control system specification : object oriented analysis and knowledge based system approach, ISPE/IFAC International Conference on CAD/CAM, Robotics and Factory of the future, Ottawa, Canada 21-24 Août 1994.

Huguet, P. Grabot, B. (1995) A conceptual framework for shopfloor production activity control", to be published in *International Journal of Computer Integrated Manufacturing*.

Hynynen, J. (1988) A Framework for Coordination in Distributed Production Management, PhD Thesis, Helsinki University of Technology.

Jones, A. Saleh, A. (1990) A multi-level/multi-layer architecture for intelligent shopfloor control. *International Journal of Computer Integrated Manufacturing*, **3**, 60-70.

Lyons, G. J. Duggan, J. Bowden, R. (1990) Project 477: Pilot implementation of a Production Activity Control (PAC) system in an electronics assembly environment. *International Journal of Computer Integrated Manufacturing*, **3**, 196-206.

Nof, S.Y. (1994) Critiquing the potential of object orientation in manufacturing, International Journal of Computer Integrated Manufacturing, vol. 7, **1**, 3-16.

Roboam, M.(1988) Modèles de référence et intégration des méthodes d'analyse pour la conception des systèmes de gestion de production, PhD thesis, University of Bordeaux I.

Rogers, P. (1992) Object-Oriented Design and Control of Intelligent Manufacturing Systems, International Conference on Object-Oriented Manufacturing Systems, Calgary, Canada, 3-6 may.

Smith, J.S. Joshi, S.B. (1992) Object-Oriented Developement of shop floor control systems for computer integrated manufacturing, International Conference on Object-Oriented Manufacturing Systems, Calgary, Canada, 3-6 may.

Tardieu, H. Rochfeld, A. Coletti, R. (1983) La méthode Merise, principes et outils, Les Editions d'organisation, Paris.

Van Der Pluym, B. (1990) Knowledge-based decision making for job-shop scheduling, International Journal of Computer Integrated Manufacturing, vol 3, **6**, 354-363.