9

Collaborative autonomous control

F. Biennier [1], J. Favrel [1], J.P. Denat [2]

1 - PRISMa, INSA de Lyon - IF 502, 69621 Villeubanne Cédex France, phone : 33 4 72 43 84 85, fax : 33 4 72 43 85 18, email : biennier@ifhpserv.insa-lyon.fr
2 - LAMII / CESALP, ESIA - BP 806, 74016 ANNECY Cedex, France, phone : 33 4 50 66 60 43, fax : 33 4 50 66 60 20, email : denat@esia.univ-savoie.fr

Abstract

Taking efficiently robustness into account in a production system involves to take perturbations as « normal events » in the control architecture. In order to reduce reaction times, the global robustness of a production system can be improved by implementing a distributed control architecture. For this purpose, we propose to design a distributed autonomous control architecture, based on behaviour based principles. Then, collaboration between elementary controllers is described with particular languages taken from real time and telecommunication fields.

Keywords

Robustness, behaviour based control, formal specifications

1 INTRODUCTION

Controlling a production system involves to co-ordinate several elementary specialised controllers as workshop control, material requirements systems (Biennier et al., 1993)... The descriptions of such a complex system can be directed

Advances in Production Management Systems N. Okino, H. Tamura & S. Fujii (Eds.)

by resources as in Craye and Gentiana (1989), tasks as in Bonetto (1987) or by cognitive point of view as in Caillau et al. (1989) or in Rasmussen and Lind (1982). Such a decentralised organisation can improve reaction abilities and consequently the global robustness of the production system. Nevertheless, collaboration between controllers must be taken into account to insure a global consistency and reduced answering times.

So we will first describe the robustness concepts and requirements before setting the basic autonomous control features and describing how these elementary controllers can collaborate. An example based on the AIPRAO flexible workshop will be presented in order to illustrate the proposed approach.

2 ROBUST CONTROL OF A MANUFACTURING SYSTEM

Manufacturing systems are managed by different controllers, each of them devoted to a particular task or resource and even divided according to different temporal horizons. Introducing robustness in manufacturing control systems aims to make them more resistant to perturbations and to improve their reaction abilities.

Roughly, robustness of a production control system can be defined as its ability to « absorb » perturbations. Perturbations can be defined as 'non predictable events'. This involves two main classes of perturbations:

- some perturbations can be identified BUT one never knows when they can happen. For example, processing activities consume tools. This can be computed and well identified. Nevertheless, it is not possible to compute the exact moment when the tool will be broken,
- some perturbations are not known a priori: they are due to main changes in the production context.

These perturbations can be integrated at different levels in the 'production decision system' :

- identified perturbations can be taken into account as normal events in the production decision system. For this purpose, these perturbations must be well described:
    * how they can be identified,
    * what are their consequences,
    * how they can be planned,
    * how they can be processed,

  then, they can be integrated in the planning process as a goal or they can also be integrated in the decision process when it is described as a finite state automaton.
- unidentified perturbations can be processed only when they occur; consequently, they are only concerned by « reactions » of the decision system. After they occur, they can be identified and added to the identified perturbations if they do not involve an important cost in the decision function.

Consequently, two kinds of robustness can be defined (figure 1) :

- On one hand, control sequences can be intrinsically robust, i.e. they are able to absorb different kinds of perturbations without affecting the global performance. This is what we call the intrinsic robustness. Of course, this kind of robustness mainly deals with predictable perturbations.
- On the other hand, control sequences have to be modified to absorb some perturbations to maintain the global performance. This is what we call dynamic robustness. This kind of robustness deals with both predictable and non identified perturbations.

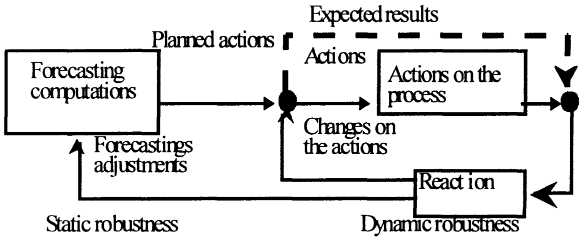


Figure 1 : Static Vs Dynamic robustness.

Intrinsic robustness involves to take perturbations as "normal" events in the planning process. So, the point consists in making an heavy use of margins and in choosing the more robust heuristic, i.e. the one that fits the best to the production context (perturbations can be included as well in the context). For this purpose, neural networks can be used (as in Guillard et al. (1991) or Rabello et al. (1993)) to classify different scheduling strategies or heuristics according to the production context. Moreover, if the scheduling process exhibits convenient performances, such a system can adapt continuously the scheduling strategy to the context. In this case, dynamic robustness is improved as well.

Dynamic robustness is quite close to reactive abilities. When a perturbation occurs, the production system must adapt its control sequence to preserve the global performance. According to this point of view, the control architecture should be as close as possible to the system. For this purpose, small autonomous controllers seems to be a convenient solution.

Communication between intrinsic robustness and dynamic robustness can be provided by a ‘ robustness supervisor’ (Biennier et al., 1995). Depending on the incoming event, the supervisor evaluates if a reaction is needed and which decision system has to react. Then these experiences are capitalised in order to improve intrinsic robustness (figure 2).

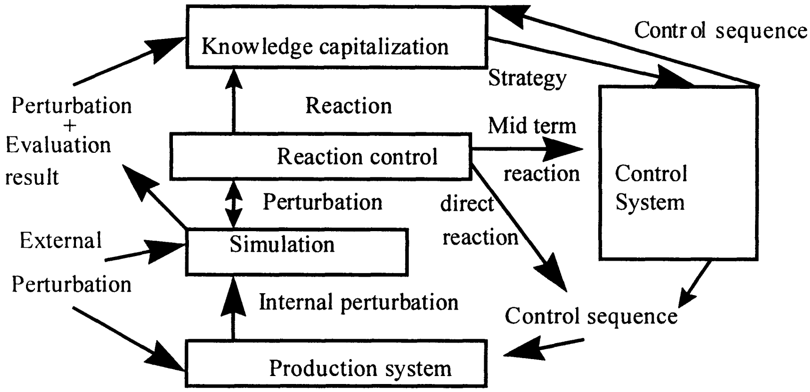


Figure 2 : Principle of the robustness supervisor.

By this way, global robustness is improved and 'on line' adaptation can be used to integrate 'unidentified perturbations' into account.

3 AUTONOMOUS CONTROL

Increasing dynamic robustness can be achieved by increasing reaction abilities. Decentralised organisation involves also autonomous controllers. For this purpose, we propose to build a self-organised system based on reactive controllers.

In 1986, R.A. Brooks proposed a new control approach called 'Behaviour based control' or 'Reactive control' (Brooks, 1986). The basic principle of this architecture is very simple: intelligent control is not directly related to the complexity of a decision centre. Since systems are qualified as 'intelligent systems' according to the behaviours they exhibit, Brooks proposed to base robots' decisions on their desired behaviour and coupled directly perception to action without using any representation of the external world.

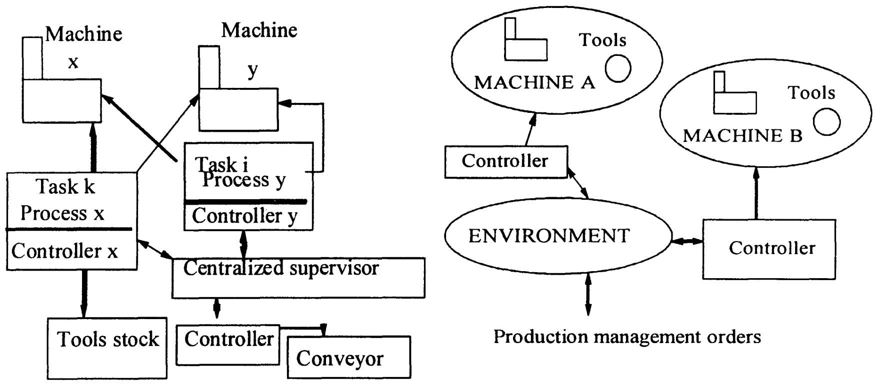
Such an approach improves self organisation and controllers exhibit robust behaviours. That is why we use this architecture to implement basic controllers for autonomous conveying robots. By this way, the decision process consists only in reactions to the environment and there is no description of the workshop. Consequently, dynamic robustness is improved. Simulations have shown that the self-organisation of this conveying system gives good results (Beslon et al., 1995) and is pertinent to implement robust control.

Nevertheless, such approaches rely totally on the graph of behaviours. When the number of modules increases, this network becomes quickly hard to build. A methodology and a conception language may help the designers to build evolved controllers but learning algorithms appear as a possible solution to build automatically complex behaviour based controllers (Maes and Brooks, 1990). Nevertheless, none of these approaches takes collaborative features into account.

4 COLLABORATIVE CONTROL

Collaborative control of a production system involves first to define the global decision system and the way it is distributed among elementary controllers. For this purpose, two main strategies can be used (figure 3):

- the decision architecture is a 'task driven architecture': in this case, a controller is defined for each task and may control different resources (a)
- the decision architecture is a 'resource driven architecture' : in this case, a controller is attached to each resource and may be used by different tasks (b).



(a) a task driven architecture (b) a resource driven architecture

Figure 3: Descriptions of a same decision process.

Co-operation between controllers can be achieved in different ways:

- directly, if the output of a controller is used as an incoming event by others; in this case, reaction times are related to the number of connected controllers between an event to the final answer,
- indirectly, if controllers communicate only thanks to the environment they share; in this case, the main problem relies on the inertia of the environment: controllers can react only when they can detect a significant event,
- at the 'knowledge level', if controllers can share a common information and knowledge system; by this way they can anticipate changes in their common environment.

The collaborative architecture we propose is a layered architecture (figure 4):

- Elementary controllers are gathered into an hypergraph structure in order to provide a communication at the knowledge level, so each elementary controller can have a global point of view on the system (Biennier et al., 1993). This feature is achieved by using epigenetic cognitive hypermaps, an extension of Kosko's cognitive maps (Kosko, 1986).
- Indirect collaboration is improved with a 'news group' mechanism. Each controller is related to a list of incoming events and in a symmetrical way, each identified event is related to a list of controllers. By this way, a controller can anticipate its reaction and processes the event BEFORE being

informed of its effects on the environment. Of course, unidentified events are processed traditionally according to the action they have on the environment. Nevertheless, in order to improve intrinsic and dynamic robustness, they can be inserted in the incoming events list just after their first processing,

- Direct collaboration is related to a graph of elementary controllers. Consequently, each controller can export some events or parts of its context to other controllers. This mechanism is similar to the import / export mechanism defined in ESTELLE (ISO/TC97/SC21/WG16-1, 1988) or SDL (CCITT, 1985). A broadcast mechanism is also available in order to provide the same information to all the controllers belonging to a communication network.

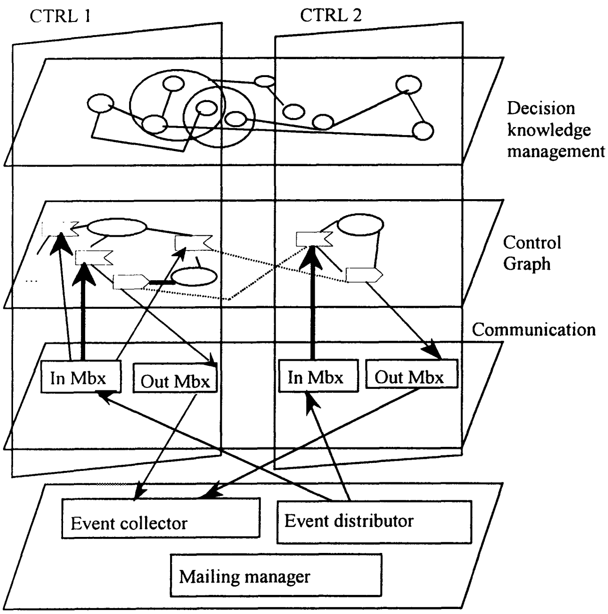


Figure 4: General organisation of the global architecture.

The global decision process can be described with complementary tools taken from real times systems as SADT methods introduced by (Ross 1977) or the LA4 system described by (Schwarz et al. 1994) and from the telecommunication fields as SDL (CCITT, 1985). By this way different abstraction levels and precise formal descriptions can be achieved in a similar way as concurrent engineering activities are described (Biennier et al. 1994).

As the communication network architecture is related to the information flows and to the time constraints, the communication graph used for the direct collaboration can be partly deduced from the network architecture. In a similar way, high level network addressing features can be used to identify the different decision systems and their hierarchy.

This communication architecture can also be taken into account in order to adapt the information flows to the communication features. For this purpose, different levels can be defined and decision functions can be physically distributed among different elementary decision and processing units (figure 5):

- **level 0** consists in field bus: it should be a low cost network, with short answering times; it connects sensors and actuators; This instrumentation level uses particular communication protocols as IEEE488 (Standard IEC 625). Fieldbus (standard EN50170) as WFip (norm UTE C40-601/46-607), Profibus or Pnet offer a good solution for this level. It allows to guarantee the coherence of the information in a multipoints network. WFip uses MMS (ISO standard IS 9506/1 and 9506/2) at OSI Application level.
- **level 1** consists in connecting machines, responsible for a set of operations; orders are given by level 2 and this level co-ordinates and processes information and tasks achieved in level 0; communication at this level uses particular software specifications with critical answering time; this is done for example in the MiniMAP standard (standard ISO 8802.4); Ethernet (standard ISO 8802.3) or TokenRing (standard ISO 8802.5) can also be used.
- **level 2** manages the workshop and has to co-ordinate a set of machines; it has to download processes on the convenient machines, control and synchronise different tasks... For this purpose tools as MMS (ISO standard IS 9506/1 and 9506/2), can be used efficiently;
- **level 3** is the main network: answering times are no more critical at this levels; it connects the different CAX systems and the communication requirements consists mainly in file transfer and control or in message interchange; consequently, traditional tools as files transfer (FTAM, FTP), shared files (NFS, Windows NT, Netware) , or E-mail (X400, SMTP).

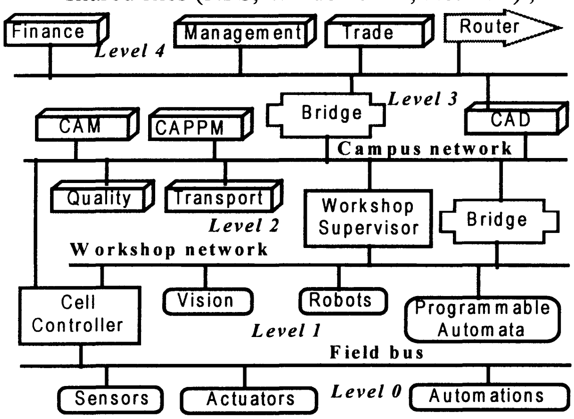


Figure 5: Organisation of different network features.

Moreover, this 'adjusted' distribution favours a modular definition of the decision process, and a convenient choice of the communication software.

5 EXAMPLE

Depending on the control organisation (a task driven approach or a resource driven approach), different architectures can be built for a same system. In the following example, we present briefly these two strategies applied to a part manufacturing in the AIPRAO flexible workshop presented figure 6.

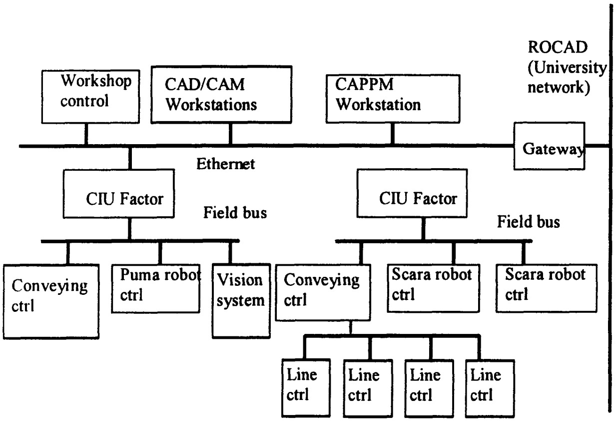
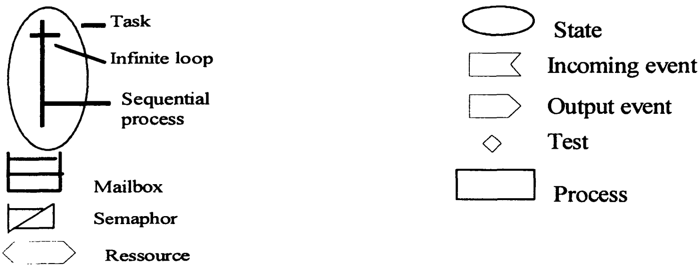


Figure 6: Synoptic of the AIPRAO physical system.

Manufacturing a part involves to co-ordinate both the conveying system, robots and machines. The global architecture for the two main strategies will be described with the LA4 and SDL graphical languages (figure 7).



(a) LA4 graphical language (b) SDL graphical language

Figure 7 : Graphical languages used

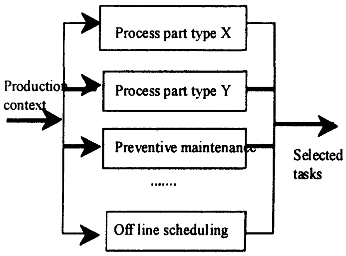
5.1 Task driven description

Such an organisation is rather linked to a 'planned organisation' after choosing a particular process, different systems are collaborating to achieve a particular process.

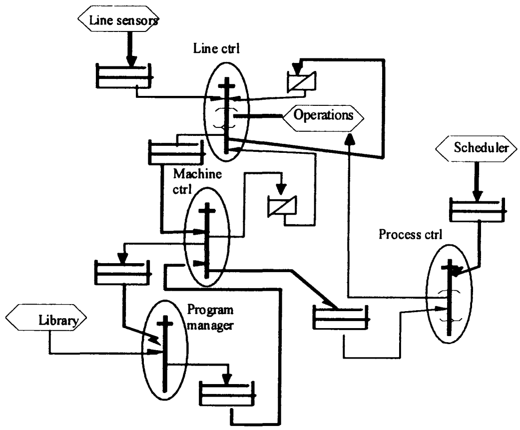
In a task driven organisation, each task can be seen as a particular behaviour. Consequently, a general graph of behaviour is linked to the global workshop in order to select the convenient control processes depending on the context (8a). Then , each elementary behaviour consists in:

- an evaluation process which computes the interest level of selecting this behaviour according to the context,
- a description of the control system associated to this particular behaviour.

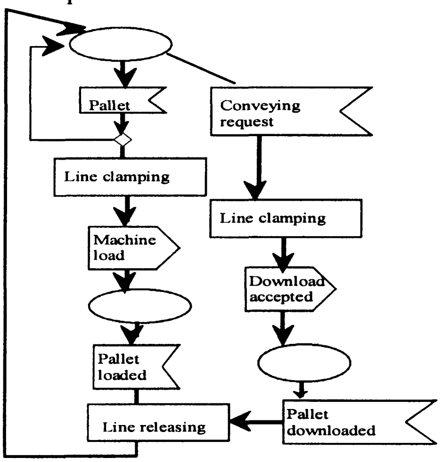
For this purpose, the different controllers MUST be synchronised. This can be formalised with a general synoptic using the LA4 language (8b). Then, different controllers are linked to the tasks and described in SDL automata (8c, 8d).



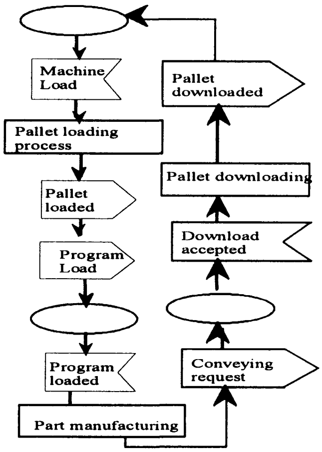
(8a) : Behaviour based control of a task driven workshop.



(8b) LA4 description of the process 'part manufacturing'.



(8c) SDL automaton of the conveying controller

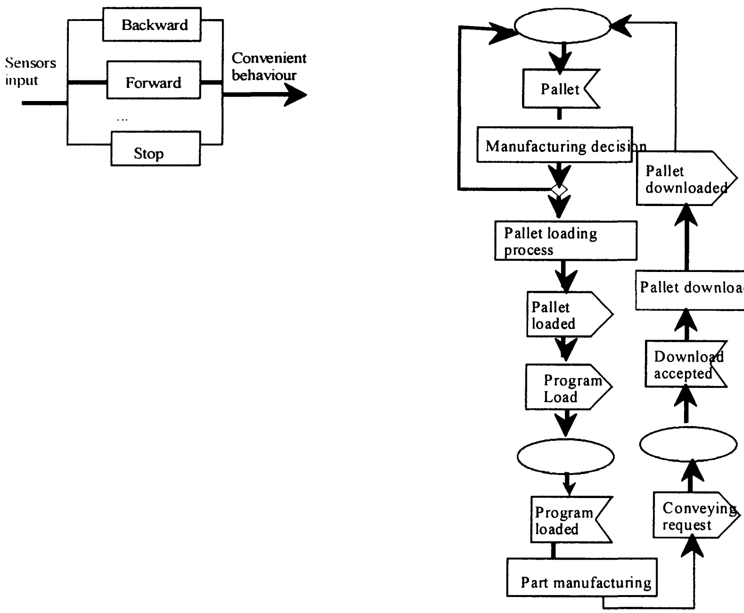


(8d) SDL automaton of a machine Controller

Figure 8 : Task driven control.

5.2 Resource driven approach

In this case, reaction abilities are developed for each controller and such an architecture is quite close to a self-organised control system. In his case, there are no general control structure: each controller is designed to take into account different events. Consequently a graph of behaviour is attached to each controller (figure 9a) shows the elementary graph associated to a line controller). Then, contextual behaviours can be described in SDL automaton (figure 9b) shows the automaton associated to a particular machine).



(a) Elementary graph of behaviour (b) SDL automaton associated to a machine

Figure 9 : Resource driven description

6 CONCLUSION

The architecture proposed in this paper is based on reaction abilities and aims to increase robustness. For this purpose, a reactive controller is proposed, based on the behaviour based control principles. By this way, self-organisation is improved. Nevertheless, such elementary controllers must collaborate in order to provide an efficient control. For this purpose, we propose a three levels architecture (figure 4), mainly based on communication features.

Simulations have shown the interest of such a reactive control. Nevertheless, few experiments are needed in order to implement directly these features with existing communication software, as MMS for example, in order to improve the response times.

7 REFERENCES

Beslon G., F. Biennier and J. Favrel (1995). A flexible conveying system based on autonomous robots. CARs and FOF'95 proceedings, Pereira (Colombie), August 1995, vol. 1, pp. 115-120

Biennier F., J. Favrel and G. Beslon (1993). Integration of information and knowledge from the engineering activity to the workshop control. IEEE International Conference on Robotics and Automation proceedings, pp 840-845, New York, IEEE Press

Biennier F., P. Coquard and J. Favrel (1994). Modelling collaborative work in concurrent engineering. Proceedings of the IFIP International Conference on Feature Modelling and Recognition in advanced CAD/CAM Systems, Paris: Hermes, pp. 125-139

Biennier F., G. Beslon and J. Favrel (1995). Robust Control For Manufacturing Systems. IMS'95 Proceedings, Bucharest (Romania), November 1995, pp. 349-354

Bonetto R. (1987). Les ateliers flexibles de production. Hermès, Paris

Brooks R.A. (1986). A robust layered control system for a mobile robot. IEEE journal of robotics and automation, vol. RA-2 (1), pp. 14-23

Caillau A., F. Meftouh, F. Donnarel and P. Lopez (1989). Driving assembly unit by means of expert system. AIPAC'89 Proceedings, vol. 2, pp. 224-228

CCITT, 1985. Red Book, vol. VI.10, "Language for Description and Functionnal Specification (SDL)", Advices n° Z.100 to Z.104, 1985

Craye E. and J.C. Gentiana (1989). Distributed implementation and hierarchical level for control of flexible manufacturing systems. AIPAC'89 Proceedings, vol. 2, pp. 190-193

Guillard S., P. Baptiste and J. Favrel (1991). Modelling and simulation for self organization in modern production workshop. CAPE'91 proceedings, pp. 705-712

ISO/TC97/SC21/WG16-1 (1988). Estelle: a formal description technique based on an extended state transition. DP9074

Kosko B. (1986). Fuzzy cognitive maps. International journal of man-machine studies, n°24, pp. 65-75

Maes P. and R.A. Brooks (1990). Learning to coordinate behaviors. AAAI proceedings, pp. 796-802

Rabello L., Y. Yih, A. Jones and J.S. Tsai (1993). Intelligent scheduling for flexible manufacturing systems. IEEE International conference on robotics and automation proceedings, vol. 3, pp. 810-815, New York, IEEE Press

Rasmussen J. and M. Lind (1982). A model of human decision making in complex systems and its use for design system control strategies. American control conference, pp. 270-276

Ross D.T. (1977). Structured analysis: a language for communicating ideas. IEEE Transactions on software engineering, vol. 3, n°1

8 ACKNOWLEDGEMENT

This work is a part of the CORINE project supported by the Région Rhone Alpes (FRANCE) and the MESR.

9 BIOGRAPHY

Frédérique Biennier is an assistant professor at the INSA de Lyon. She received her engineering degree in computer science in 1988 and the Ph. D. in computer science and automatics in 1990. Her main teaching activities concern neural networks theory and practice, hardware features, real time systems, telecommunication systems and services and production management. Her main research topics are production control (scheduling activities, workshop control) and designing support systems for concurrent engineering (information systems, collaborative work management) thanks to common tools as neural networks or formal specifications.

Joël Favrel is a professor at the INSA de Lyon. He received his engineering degree in electronics at the INSA de Lyon in 1964 and his doctorate of Science at the University Claude Bernard - Lyon Science in 1968. His main teaching activities concern probability and statistics and Computer Integrated Manufacturing. His main research areas are related to CIM and specially scheduling, group technology and information systems for production. He is currently in charge of the production engineering research group at the LISPI laboratory, and of an experimental CIM centre (AIP RAO). He is also a French representative at IFIP in the working group 5.7 (Computer Aided Planning).

Jean Paul Denat is an assistant professor at the ESIA, Université de Savoie. He received his Ph D in applied physics in 1976. Since 1985 his research and teaching activities have moved to discrete events systems (finite state machines, Grafcet, Petri nets...). His research areas are related to robust control of discrete systems with staying time constraints.