

An Approach to the Formal Specification of Holonic Control Systems

Paulo Leitão¹, Armando W. Colombo², and Francisco Restivo³

¹ Polytechnic Institute of Bragança, Quinta Santa Apolónia, Apartado 134,
P-5301-857 Bragança, Portugal,
`pleitao@ipb.pt`

² Schneider Electric GmbH,
Steinheimer Str. 117, 63500 Seligenstadt, Germany,
`armando.colombo@modicon.com`

³ Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias,
P-4200-465 Porto, Portugal,
`fjr@fe.up.pt`

Abstract. In the manufacturing world, globalisation leads to a trend towards the reduction of batches and product life cycle, and the increase of part diversity, which are in conflict with other requirements, such as the cost reduction achieved with higher productivity. Thus, the challenge is to develop flexible, agile and intelligent management and control architectures that satisfy the referred requirements. The holonic manufacturing and the agent-based manufacturing approaches allow a new approach to the manufacturing problem, through concepts such as modularity, decentralisation, autonomy and re-use of control software components. ADACOR, one of the holonic architectures recently proposed, defines a set of autonomous and intelligent holons aiming to improve the performance of control system in industrial scenarios characterised by the frequent occurrence of unexpected disturbances. The formal modeling and validation of the specifications of the ADACOR-holons and of the interactions between these holons to implement the manufacturing control functions is of critical importance. In this paper, a formal methodology is introduced and applied to model the dynamic behaviour of the ADACOR-holon classes.

1 Introduction

The emergent requirements of global markets are leading the manufacturing world to the reduction of batches and product life cycle, and the increase of part diversity, which are in conflict with other important requirements, such as the reduction of costs, achieved normally with higher productivity. A Flexible Manufacturing System (FMS) aims to fill the gap between the mass production, which guarantees high productivity, and the dedicated NC machine production, which guarantees high flexibility and customisation. A FMS is a production structure that comprises a set of workstations, such as machine tools, storage facilities,

and robots, interconnected by a transport and materials handling system, and controlled by a supervisory control system.

The flexibility and performance of FMSs depend not only of the individual components, i.e. workstations, storage facilities, etc., but also of the flexibility and performance of the embedded control system [1]. The manufacturing control systems are concerned with coordinating the manufacturing resources to make the desired products. The main functions presented in the manufacturing control system can be split into: process related functions (process planning) and resource allocation related functions. The resource allocation comprises the following main functions: resource allocation planning (scheduling), resource allocation plan execution (dispatching, monitoring, diagnosis, reaction to disturbances, etc.) and pathological state handling (deadlock handling, etc.). The scheduling determines an optimal route with respect to some performance criteria, and the plan execution performs the final assignment of resources to the orders, based on the actual state of the manufacturing system and the schedule plans. The pathological state handling intends to keep the system in a safe state and/or recovers it from undesirable states.

The manufacturing systems are typically heterogeneous environments, comprising heterogeneous hardware components and software applications, with distributed functions, knowledge and skills, which are required to cooperate in order to achieve common goals. The control system should be therefore be based in distributed and autonomous entities, expandable, being possible the addition of new components without the need of re-design, re-programming and re-initialisation of the other components, and re-configurable, adapting dynamically to configuration changes, without stopping or re-starting the process. Additionally, the manufacturing systems are complex non-linear systems, since the occurrence of a disturbance causes non-linear impact in the system. For this reason, their occurrence may have severe impact in the performance of manufacturing systems, being also necessary to improve the system performance in terms of response to change. All these necessary facilities lead to the concept of agile manufacturing systems.

The challenge is to develop new flexible, agile and intelligent management and control architectures that address the above referred problems and requirements. The holonic manufacturing and the agent-based manufacturing approaches that have been introduced in the manufacturing domain by several research teams, such as referred in [2–7], allow a new approach to the manufacturing problem, through the concepts of modularity, decentralization, autonomy and re-use of control software components. One of the holonic architectures proposed during the last two years is the ADACOR (Adaptive Holonic Control Architecture for Distributed Manufacturing Systems) architecture [8], which defines a set of autonomous, self-organised and intelligent holons in order to improve the performance of control system in industrial stochastic scenarios, characterised by the frequent occurrence of unexpected disturbances.

The formal modeling and validation of the structural and behavioural specifications of the ADACOR-holons and the interactions between these holons to

implement the manufacturing control functions assumes critical importance. In this paper, a formal methodology to model the behaviour of the ADACOR-holon classes is discussed and applied.

This paper is organised as follows: First, Section 2 discusses the need for a formal methodology to model the specifications of holonic control systems. In Section 3 it is described the specifications and modelled the behaviour of ADACOR-holon classes using Petri Nets (PN) modelling tool. Finally, Section 4 rounds up the paper with conclusions and an overview of planned further developments related to the approach presented here.

2 Formal Methodology

In order to formalise the structure and the behaviour of the holonic manufacturing control systems, and to validate its behaviours and particularity to analyse the co-operation and interaction between the distributed holons, aiming to understand and synthesise the structure and behaviour of the system, it is important to count with a formal modelling methodology.

The proposed methodology for the formal modelling of holonic applications, as illustrated in Fig. 1, combines the UML (Unified Modelling Language) and the PN modelling tools. UML [9] is an object oriented based modelling tool, that is adequate to model the structure and the static aspects of a manufacturing system. In the proposed formal methodology, the static aspects are modelled using mainly the class diagrams, which shows the classes of objects in the system, the attributes and methods for each class, and the relationships between the objects.

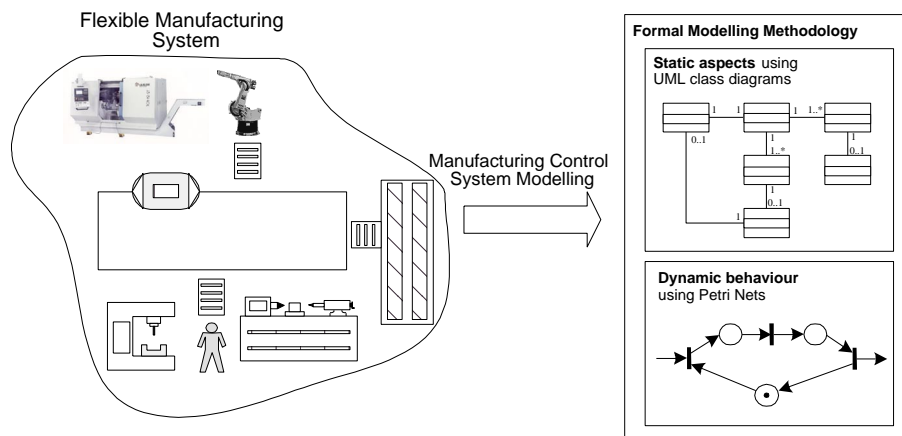


Fig. 1. Modeling a Manufacturing Control System

The modelling of the dynamic behaviour of the system requires a formal tool that captures characteristics like concurrency or parallelism, asynchronous operations, deadlocks, conflicts or resource sharing, which are inherent to FMS [10]. Additionally, it is crucial that the formal modelling tool has the capability to validate the behavioral characteristics of these event-driven systems, as also the analysis of other important aspects, such as the deadlock detection and the performance analysis.

The UML modelling tool doesn't support efficiently the modeling of the dynamic behaviour aspects and the formal validation of these specifications. On the other hand, the PNs is a formal modeling tool, both graphical and mathematical, that seems adequate to model and analyse the structure and the dynamic behaviour of complex event-driven systems with high distribution degree. In comparison with UML, the PN formalism allows to design the control system behaviour, but also to validate and verify the behaviour of the system, based in mathematical background embedded in the PN formalism. In this sense, the proposed methodology uses the PN formalism to model the dynamic behaviour of the holonic manufacturing control system. More details about PN theory and mathematical fundamentals are out of the scope of this work. We recommend the readers to consult the following references [4, 12, 13].

In industrial manufacturing applications, the PN models become highly complex and difficult to handle. This leads to the definition and application of different types of High-Level Petri Nets, depending on the area of application, i.e., modelling and qualitative analysis, quantitative/performance analysis, modelling of big complex manufacturing environments, supervisory control code generation, etc. [1, 14, 15].

In the following, a kind of PN tailored for production management and control modelling purposes, proposed in [12], will be used to model the dynamic behaviour of the different holon-types defined in ADACOR holonic architecture, allowing to get a comprehensive formal view of the structure and behaviour of these holon-types.

3 Modelling the Holons Dynamic Behaviour in ADACOR Architecture

The ADACOR architecture proposes a holonic approach to introduce the dynamic and agile adaptation to disturbances in flexible manufacturing systems [8]. Aiming to support the distribution of skills and knowledge, the architecture is based on a set of autonomous, intelligent and co-operative entities, designated by holons, to represent the factory components. These distributed components can be both physical resources (numerical control machines, robots, programmable controllers, etc.) and logic entities (products, orders, etc.). According to the generalization concept of the object-oriented paradigm, the ADACOR architecture groups the manufacturing holons into product, task, operational and supervisor holon classes [16].

The dynamics and behaviour evolution of a manufacturing control system will be modelled through the modeling of dynamic behaviour of each individual holon class in the system using PNs, with the places representing the state of the holons when executing activities and the transitions representing the trigger of actions and the synchronization between holons or between threads within a holon. The tokens in those PN models can represent resource states, parts in the system or logical control.

3.1 Product Holon Model

Each product is represented by a *product holon* that contains all knowledge related to the product and is responsible for the process planning. The product holon receives orders to execute products, which can be customer orders from customer entities or forecast orders based in historic information in case of production to stock. To execute the products, the product holon require the information about the BOM (Bill of Materials) that specifies the product structure, and the process plan required to execute the product, that should be provided in the product data model, created by the engineering department during the product design phase.

In functional terms, which PN-model is illustrated in Fig. 2, the product holon start its execution entering in a state waiting for new product orders. These new orders will generate a new thread to handle the execution of each order, continuing the product holon waiting for new orders, being able to process simultaneously several product orders.

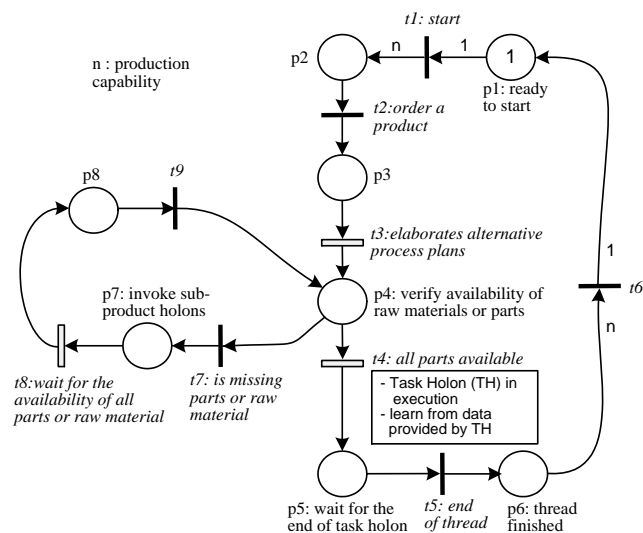


Fig. 2. Product Holon Behaviour Model

In each thread, initially it is elaborated several alternative plans for the execution of the product, based in the knowledge related to the operations routing and in the available resources in the system, indicating for each operation a set of alternative resources to execute it, sorted by the confidence degree on concrete resource to execute it. Then it is verified if the required parts or raw material are available on the storage system, interacting with the operational holon responsible for the storage management. If the raw materials or parts are not available, the product holon interact with other product holons to request the execution of sub-products, according the product structure. When all sub-products or raw materials are available it launches a task holon that will be responsible for the supervision of the manufacturing order leading to the product execution.

After launching a new task holon to deal with the execution of the manufacturing order, the thread will wait for the conclusion of the manufacturing order. At this moment, the task holon notifies the product holon providing the relevant information concerning the execution of the part (for example the process plan used, the start and end dates, etc.). These data should be carefully analysed, allowing from the execution of the manufacturing order, to learn to elaborate more efficient and accurate process plans. The process plans generated in the future will take into consideration penalties to the resources that in previous operations had failures, delays or operations with low quality, and rewards to the resources that executed with success previous operations.

3.2 Task Holon Model

Each available manufacturing order is represented by a *task holon*, which is responsible for the control and supervision of the manufacturing order execution and contains the dynamic information. The task holon functions comprise the order decomposition, resource allocation planning and resource allocation plan execution, as illustrated in the PN-model of Fig 3.

Initially, the task holon requests a pallet and material to the transport and storage system. According to the production type, the pallet may contain several parts of the same material or contain all necessary parts to execute the final part.

In the resource allocation process, the task holon announces the operations belonging to the manufacturing order, by interacting with available supervisor and operational holons, deciding the allocation of each operation according the bids received.

After allocating all operations, the task holon will start the execution of operations, interacting with the operational holons, even if the allocation is provided by the supervisor holons. For each operation it is necessary to prepare the operation to be executed. This preparation involves the transportation of the parts to the machine where the parts will be processed, and the execution of set-ups in the machine, if necessary. Before requesting the execution of transport of the part to the machine, the task holon should ask if the machine can accept to receive the part that will be transported (for example, to avoid deadlocks due to no space in the machine buffer).

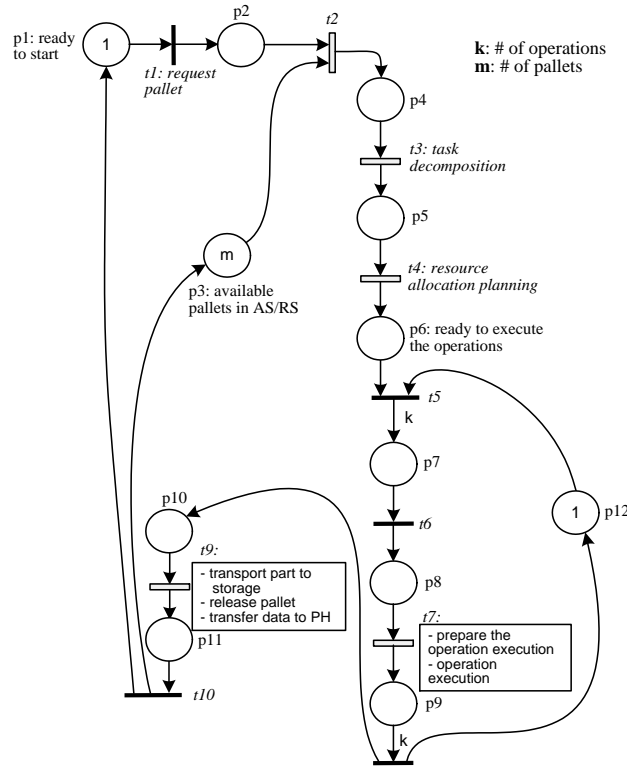


Fig. 3. Task Holon Behaviour Model

When both processes, the transportation of the part and the execution of the set-up, are completed, the task holon can start the execution of the operation, notifying the operational holon. Once the operation is started, the control is given to the operational holon, and the task holon waits for the end of the operation. The described procedure for one operation is repeated for all operations that belong to the process plan of the manufacturing order.

After the execution of all operations, the task holon request the transportation of the part to the storage system and releases the pallet. The task holon finishes its execution transferring to the product holon, the relevant information about the product execution, such as the start and end dates.

3.3 Operational Holon Model

The *operational holons* represent the physical manufacturing resources, such as operators, robots and numerical control machines, managing its behaviour according the resource goals, constraints and skills, and optimising its schedule agenda, Fig. 4.

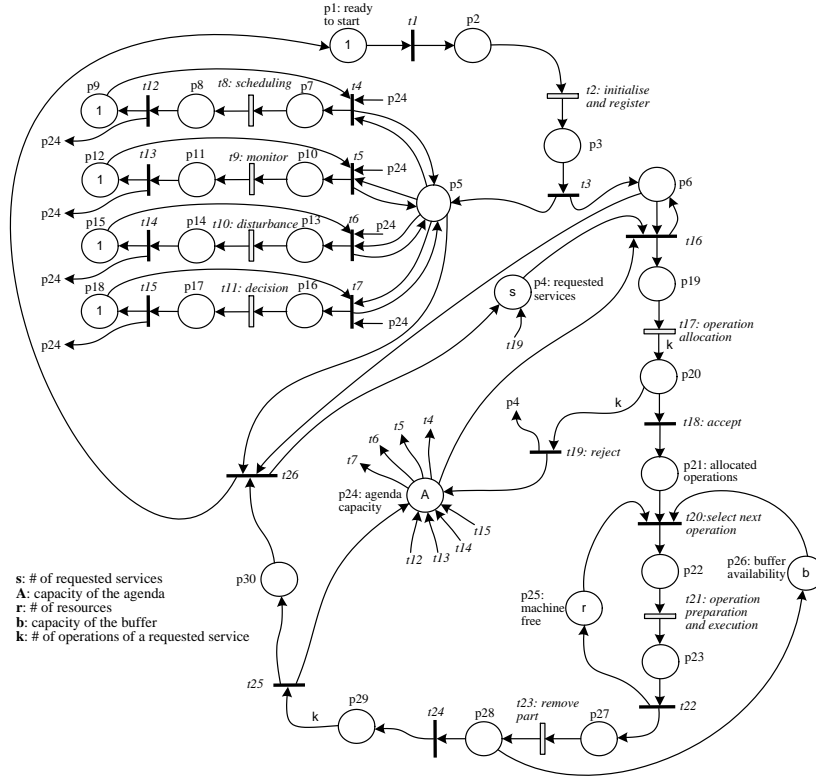


Fig. 4. Operational Holon Behaviour Model

After initialising its components and registering its capacity and skills in the appropriated supervisor holon, according to the organisational structure, the operational holon behaviour acts as a reactive server, in the sense that it is waiting for new operations (proposed by the supervisor holon or by the task holons) and it has the possibility to execute monitoring, scheduling and disturbances handling.

The sub-behaviours are handled asynchronously using threads, so that the execution of one process doesn't block the execution of another process; for example, when monitoring the execution of an operation, the operational holon can handle the announcement of new operations or execute scheduling. The monitoring, disturbance handling and scheduling activities are performed concurrently.

The operation allocation is analysed, the acceptance of the operation allocation being decided according the autonomy factor and the actual agenda capacity. In case of acceptance, the operation is stored in the agenda, waiting for the appropriate moment to start the operation execution. According to the availability of the buffer and the state of the machine, the next operation is selected, based in the local scheduling.

The preparation of the execution involves the transportation of the part to the machine and the execution of a set-up if necessary. The set-up aims to endow the machine with the required tools and fixtures to execute the operation, and in case of need to execute a set-up, the operational holon deals directly with the operational holon that represents the team that executes the set-ups. After this phase, it is started the execution of the operation.

When the operation finishes, the resource returns to the idle state being able to initiate the execution of another operation, and the part is removed from the machine to the next machine, according to the resource allocation plan.

3.4 Supervisor Holon Model

The product, task and operational holons are quite similar to the product, order and resource holons, presented at the PROSA reference architecture [2]. The *supervision holon* presents different characteristics from the staff holons defined in PROSA, introducing coordination and global optimisation in decentralised control approaches, coordinating several operational and supervisor holons. In normal operation, the supervisor holon coordinates the activity of the holons under its domain, while when a disturbance occurs, these holons may have to find their way without the help of the supervisor holon.

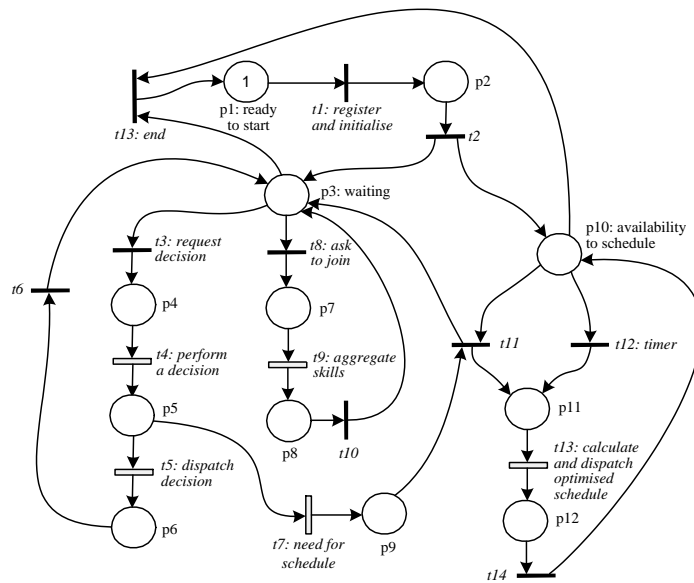


Fig. 5. Supervisor Holon Behaviour Model

The supervisor holon is also responsible for the group formation and their dynamic evolution according the environment context, based in pre-defined clus-

ters of holons, combining synergies, aggregating skills and offering the combined services to external entities in the manufacturing system. These groups can be formed to build a shop floor, a manufacturing cell, or a machine equipped with a set of tools, assuming the supervisor holon the role to coordinate each group.

The behaviour model of the supervisor holon acts simultaneously as a server and a client: as a server waiting for requests and as a client generating optimised schedules that are sent to the operational holons, as illustrated in Fig. 5.

The request to elaborate optimised schedules can be generated from the announcement of an operation, from the need to optimise the actual schedule after the end or disruption of an operation, or from the need for re-scheduling due to a disturbance. After achieving a schedule, the new scheduled operations, and the allocated operations that had modifications in schedule parameters due to the new schedule, are proposed to the appropriate operational holons and to the task holons.

The decision-making activity is related to the actions associated to monitoring, scheduling and disturbance handling, presenting more complexity than to the one presented at operational holons due to the need to handle with aggregated lower holons knowledge and skills. As the supervisor holon co-ordinates several operational and/or supervisor holons, it manages the group of holons under its coordination domain, aggregating the skills and capacity of the operational holons, when they join to the group.

4 Conclusions

The flexible manufacturing systems are complex and stochastic environments requiring the development of flexible, agile and intelligent management and control architectures that support the small batches, product diversity, high quality and low costs imposed by global markets. One of these intelligent management and control architectures is ADACOR.

The ADACOR holonic architecture aims to improve the performance of control system in scenarios characterised by the frequent occurrence of unexpected disturbances, defining four main holon classes: product, task, operational and supervisor holons. Each of these holons presents characteristics of autonomy, cooperation and intelligence, allowing to implement an intelligent distributed control system.

In this paper, it is used a kind of Petri net, tailored for production management and control modelling purposes, as a formal methodology to model the behaviour of the ADACOR-holon classes in a bottom-up approach. The individual model of each holon uses special temporised transitions to model activities execution, that can be exploded into a more detailed and refined level. These sub-models, according the degree of refinement, are the different software control modules of the hardware, i.e., a formal representation of the holons.

The edition, simulation, qualitative and quantitative (performance) analysis, and formal validation of the structural and behavioural specifications of the

ADACOR-holons and their interactions, is one of the complementary works to the approach proposed here and it can be found in [17].

A brief overview of the latest works published in the area (see e.g. [18,19]) allows to identify a set of weak points in using the PN-formalism proposed here and other similar extensions of this tool. This is particularly true if the system presents many instances of the same component (e.g., n resources need n operational Holons). In this case, the model will be increased (structure and components) in a non-controllable manner. In our opinion, the use of High-Level PN, such as those proposed in [20], allows to reduce this complexity, by compressing the representation of states, actions and events, to overcome the identified limitations and to support more complex and bigger coordination scenarios.

References

1. Colombo, A.W.: Integration of High-Level Petri Net-based Formal Methods for the Supervision of Flexible Production Systems. Tutorial Lecture at the 1st Online Symposium for Electronics Engineers, February 20th (2001).
2. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, P.: Reference Architecture for Holonic Manufacturing Systems: PROSA. In: *Computers In Industry*, 37, (1998) 255-274.
3. Fisher, K.: Agent-Based Design of Holonic Manufacturing Systems. In: *Journal of Robotics and Autonomous Systems*, Elsevier Science B.V., 27 (1999) 3-13.
4. Maturana, F. and Norrie, D.: Multi-Agent Mediator Architecture for Distributed Manufacturing. In: *Journal of Intelligent Manufacturing*, 7 (1996) 257-270.
5. Van Dyke Parunak, H., Baker, A. and Clark, S.: The AARIA Agent Architecture: An Example of Requirements-Driven Agent-Based System Design. In: *Proceedings of 1st International Conference on Autonomous Agents*, (1998) 482-483.
6. Brennan, R., Fletcher, M. and Norrie, D.: An Agent-based Approach to Reconfiguration of Real-Time Distributed Control Systems. In: *IEEE Transactions on Robotics and Automation*, 18(4), (2002) 444-451.
7. Colombo, A.W., Neubert, R. and Schoop, R.: A Solution to Holonic Control Systems. In: *Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation*, Sophia/Nice, France, (2001) 489-498.
8. Leitão, P. Restivo, F.: Adaptive Production Control Systems. In: *Proceedings of special session on Agent-based Intelligent Automation and Holonic Control Systems of the 28th Annual Conference of the IEEE Industrial Electronics Society*, Sevilla, Spain, 5-8 November (2002) 2968-2973.
9. Rumbaugh, J., Jacobson, I. and Booch, G.: *The Unified Modeling Language Reference Manual*. Addison-Wesley, (1998).
10. Silva, M. and Valette, R.: *Petri Nets and Flexible Manufacturing*. In: *Advances in Petri Nets*, Lectures Note in Computer Science, vol. 424, Springer Verlag, (1989) 374-417.
11. Murata, T.: *Petri Nets: Properties, Analysis and Applications*. In: *Proceedings of the IEEE*, 77(4), (1989) 541-580.
12. Colombo, A.W. and Carelli, R.: *Petri Nets for Designing Manufacturing Systems*. In: *Computer-Assisted Management and Control of Manufacturing Systems*, cap. 11, S.G. Tzafestas (ed.), Springer-Verlag (1997).
13. Desrochers, A. and Al-Jaar, R.: *Applications of Petri Nets in Manufacturing Systems-Modeling, Control and Performance Analysis*. IEEE Press (1995).

14. Feldmann, K., Schnur, C. and Colombo, A.W.: Modularized, Distributed Real-time Control of Flexible Production Cells, using Petri Nets. In: CEP, 4(8), (1996) 1067-1078.
15. Holloway, L., Krogh, B. and Giua, A.: A Survey of Petri net Methods for Controlled Discrete-Event Systems. In: Discrete-Event Systems: Theory and Applications, 7(2), Kluwer Academics, (1997) 151-190.
16. Leitão, P. and Restivo, F.: Identification of ADACOR Holons for Manufacturing Control. In: Proceedings of 7th IFAC Workshop on Intelligent Manufacturing Systems, Budapest, Hungary, 6-8 April (2003) 109-114.
17. Leitão, P., Colombo, A.W. and Restivo, F.: A Formal Validation Approach for Holonic Control System Specifications. Submitted to 9th IEEE International Conference on Emerging Technologies and Factory Automation, Lisboa, Portugal, (2003).
18. Vyatkin, V., Hanisch, H-M. and Ivanov, G.: Application of Formal Methods for Deep Testing of Controllers in Holonic Systems. In: Proceedings of the 1st IEEE International Conference on Information Technology in Mechatronics (ITM2001), Istanbul, Turkey, 1-3 October (2001) 53-58.
19. Frey, G., Minas, M. and John, K.: Steuerungsentwurf mit Petrinetzen. SPS-Magazin, Verlag Marburg, 4/5 (2002) 44-47.
20. Colombo, A. W., Neubert, R., and Süßmann, B.: A Colored Petri Net based Approach Towards a Formal Specification of Agent-Controlled Production Systems. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Tunisia, (2002).