

An Approach for Reconfigurable Automated Processes in Agile Manufacturing

Marco Antonio Buseti de Paula

Pontifícia Universidade Católica do Paraná – PUCPR,
Curitiba, PR, Brazil
E-mail: marco.buseti@pucpr.br

Eduardo Alves Portela Santos

Pontifícia Universidade Católica do Paraná – PUCPR,
Curitiba, PR, Brazil
E-mail: eduardo.portela@pucpr.br

Abstract

This paper presents a project methodology applied to reconfigurable processes in the context of agile manufacturing systems. The proposal methodology consists of a cyclic three-stage development – modeling, synthesis and implementation – until the real system accomplishes the required application, resulting in the project of the automated and integrated system. The present paper details the three stages and describes which steps must be executed on each one of it. Also, the mathematical formalism used in methodology is presented, as a basis for implementation of reconfigurable processes. To submit to a test and validate the proposal approach, the methodology is applied to a manufacturing system's prototype. In the experiment, it is considered that 2 products must be manufactured in different times, demanding the processes can be reconfigurable in a fast and reliable way.

Keywords: reconfigurable processes, agile manufacturing, design methodology, systems integration, discrete event systems, manufacturing technologies

Introduction

In order to be competitive, manufacturing, more than any other activity area in economy, needs to continuously adapt to changes in the market. The increase in global competition is pushing enterprises to reduce the time response when launching new products and to offer competitive prices. Diversity, fluctuations in demand, the short life cycle of products due to the frequent introduction of new needs, in addition to the increase in the client's expectations in terms of quality and delivery time, are nowadays the main challenges which companies have to deal in order to keep competitiveness and stay in the market.

The concept of Agile Manufacturing (AM) was first introduced in the early 90's as a response to the constant changes in the new economy and as a basis for competition in a globalized economy (Wang et al., 2005a,b). Based upon this general concept, Sanchez and Nagi (2001) point out that the meaning of AM depends on the company's segment and its context. According to the authors, nine categories can be defined through the identification of papers related to AM, such as product design and manufacturing systems, supply chain, information systems and process planning.

According to Kusiak and Salustri (2007), Kusiak and Huang (1996) and Chun-Che and Kusiak (1998), AM can be reached through the strategy of product modularisation or the adoption of reconfigurable processes. The authors state that product modularity consists of using common units with the purpose of creating product variants. The advantage with this approach is to allow a scale economy, increase product variety, decrease lead-time, increase product/component change and have easy diagnoses, maintenance, and repair and product discharge (Kelkar et al., 2005). According to Rao et al. (2006) and Zhang et al. (2002), the reconfiguration of manufacturing systems can be accomplished through hardware changes (layout, machine updating and computer hardware) and software (integrated control of hardware components). The software reconfiguration consists of quickly and efficiently creating and modifying the control system. These reconfigurations allow greater flexibility to the manufacturing system, thus making it agile (Hoda and Maraghy, 2006; Bi et al., 2007). According to Gamberi et al. (2008), Manzini et al. (2005) and Manzini et al. (2004), there are two kinds of flexibility: Capability flexibility, ability of a system to adapt to changing market demands in terms of product variations and changes; Capacity flexibility, ability of a system to react to changing market demands in terms of product quantities.

Wadhawa et al. (2005) and Moore et al. (2003) add that the system flexibility is related to a control system's implementation time. Generally, the system flexibility imposes that the implementation time for new applications, which can demand the reconfiguration of software and hardware, be as short as possible, since new products invariably have new requirements and specifications related to automation, layout and integration. However, Ollero et al. (2006) and Erbe (2002) point towards the insufficient flexibility in highly automated manufacturing systems. Furthermore, the authors identify that the main reason derives from a combination of losses in consequence of this conversion, i.e., the lead-time of equipment and the high cost of specialized maintenance, which restricts the expected profits. Therefore, many factories are decreasing the level of automation in the shop floor, or are planning to do so.

In order to approach the flexibility issue in the context of reconfigurable processes, it is essential to establish systematic procedures that may characterize the development cycle of a control system (Goyon et al., 2004). This systematisation consists of using formal models for analysis, synthesis and implementation of control systems for Discrete Event

Systems (DES). A DES can be defined as a dynamic system that evolves according to the occurrence of events. A classical example of DES is a manufacturing system. Cassandras and Lafortune (2008) enumerate the main models that are used: Timed Petri Nets, Controlled Petri Nets, Markov Chains, Queuing Theory, Generalized Semi-Markov Process and Simulation, Max-Plus Algebra, Formal Languages and Automata.

Among the models previously mentioned, the Controlled Petri Nets (Holloway et al., 1997) and the Supervisory Control Theory (SCT) (Ramadge and Wonham, 1989), based on Formal Languages and Automata (Carrol and Long, 1989), are more appropriate for the development of control systems. Differently from the other models, which emphasize the systems analyses, the two mentioned models are provided with procedures for the synthesis of control systems.

The model proposed by Ramadge and Wonham (1989) allows an automatic control synthesis process instead of the usual manual and heuristic procedures. In addition to this advantage, the synthesis procedure demands that the obtained supervisor must always fulfill the control specifications. This way, new control systems may be rapidly and automatically designed when modifications, such as redefinition of specifications and physical changes, are necessary. For these reasons, the present work uses the SCT as a formal tool to obtain the supervisors for the manufacturing automated systems.

This work presents an approach for the project and for the development of the control system in a context of reconfigurable processes, aiming at an agile manufacturing. The contribution of the proposed approach, characterized by a development cycle – modeling, synthesis and implementation –, consists of treating the manufacturing automated systems project with higher efficiency, effectiveness and reliability when new applications are necessary. Such applications might come from the insertion of new products, from the reconfiguration of processes and existing products, from new demanding necessities or from technological modernization. To do that, the present paper proposes a hardware and software environment in order to implements the mentioned approach.

The aim of this paper is to provide an approach in which the automated and integrated systems' designers can reconfigurable the manufacturing processes whenever a new demand (application) has been established. For reach this, the paper aims to integrate formal tools (SCT), software and hardware platforms in order to establish the methodology.

The present paper is organized as follow. Section 2 presents briefly a conceptual review of the mathematical formalism used, which involves the modelling and control of manufacturing systems, as well as the associated implementation techniques. Section 3 presents the proposed approach, giving details of the stages that compose it. The example of application is shown in Section 4. As an application test, the development cycle is used to reconfigure the process of a new manufacturing system prototype due to the necessity of including a new product (specification changes). In Section 5 is presented the conclusions and future works that can be derived from the proposal approach.

Modeling and Control of Manufacturing Systems

In the SCT, the plant is represented by an automaton G , being $G = (\Sigma, Q, \delta, q_0, Q_m)$, where Σ is the alphabet of events; Q is the set of states; $\delta: Q \times \Sigma \rightarrow Q$ is the state transition function; $q_0 \in Q$ is the initial state and $Q_m \subseteq Q$ is the set of marked states. Marking of states allows specifying the system's completed tasks. The set of events is partitioned into two disjoint sets, being the set of controllable events $\Sigma_c \subseteq \Sigma$, and the set of uncontrollable events $\Sigma_u \subseteq \Sigma$. An event is classified as controllable if its occurrence can be disabled by the control action of some supervisor. It is classified as uncontrollable in the opposite case. The plant is associated with two languages: the generated language $L(G)$ representing all the possible sequences of events in G ; and the marked language $Lm(G)$ which represents the accomplished tasks or the marked behavior of G .

A supervisor $S: L(G) \rightarrow 2$ is a function that maps from the sequence of generated events to a subset of controllable events to be enabled or disabled. The optimal behaviour of the plant G under supervision of supervisor S is represented by the language marked by the automaton S/G . The necessary and sufficient conditions for the existence of a supervisor are presented in Ramadge and Wonham (1989). A possible representation of a supervisor is a pair $S = (S, \Phi)$, where $S = (Q^s, \Sigma^s, \delta^s, q_0^s, Q_m^s)$ is an automaton with $\Sigma^s = \Sigma^G = \Sigma$ and $\Phi: Q^s \rightarrow 2$ is an output map that specifies the subset of controllable events to be disabled at each state of the automaton representing the supervisor. Usually the automaton representing the supervisor is the automaton S/G itself. Minhas (2002) and Su and Wonham (2004) deal with the reduction of supervisors. The reduction of the supervisor S is the achievement of another representation of it, namely $S = (Sr, \Phi_r)$, where the automaton Sr has a smaller number of states than the automaton S/G and such that this reduction does not affect the control action of the supervisor.

According to the Local Modular Control (LMC) approach [26], the system to be controlled is modeled by a Product System Representation (PSR), *i.e.*, by a set of asynchronous subsystems $\{G_i \mid i \in I\}$ such that all pairs of subsystems in this set have disjoint alphabets (Ramadge and Wonham, 1989). The behavior of each subsystem is represented by an automaton $G_i = (\Sigma^{G_i}, Q^{G_i}, \delta^{G_i}, q_0^{G_i}, Q_m^{G_i})$, such that the behavior of the entire system to be controlled is obtained by the synchronous product [31] of all subsystems of the PSR, *i.e.* $G = \prod_{i \in I} G_i$. The whole set of events is $\Sigma = \cup_{i \in I} \Sigma^{G_i}$. Considering a subsystem in $\{G_i \mid i \in I\}$, $\Sigma_c^{G_i}$ denotes its set of controllable events and $\Sigma_u^{G_i}$ its set of uncontrollable events.

The LMC approach states that, instead of synthesizing a single global supervisor that satisfies the entire set of specifications, one local supervisor is synthesized in order to satisfy each specification. Each one of the local supervisors restricts the behaviour of a part of the system to be controlled. This part is the local plant corresponding to the considered supervisor. A local plant (G_l) is obtained performing the synchronous product of the subsystems which share events with the considered specification. The synthesis of

a local supervisor (S_j) is performed considering the corresponding specification (E_j) and its local plant (G_j). By using this procedure, it is possible to synthesize a local supervisor for each one of the established specifications. If at least one local supervisor in the set $\{S_j \mid j \in J\}$ disables the occurrence of an event, then the occurrence of this event is disabled in G . Even when each local supervisor is non-blocking the concurrent control action of the whole set of supervisors may result in the blocking of the entire system. Therefore, after accomplishing the synthesis procedure, it is necessary to verify the modularity property of the set of supervisors as stated in Queiroz and Cury (2000).

With the purpose of executing the local modular control approach in a readable structure, we propose programming the control system in a three-level hierarchy (Queiroz and Cury, 2002b), according to Figure 1. The set of local modular supervisors is implemented in this level exactly as theoretically conceived by Ramadge and Wonham (1989). The program updates the active states according to the automata's structures and to the state changes in the Product System level. A feedback map associates the active states to a set of disabling signals, which control the Product System. The Product System level's main function is to execute the commands that are allowed by the plant and are not disabled by the supervisors. The parallel evolution of the asynchronous sub-plants follows executed commands and responses from the Operational Sequences level, which signal state changes to the controllers.

Operational Sequences work as an interface between the theoretical Product System and the Real System. At this level, the program interprets the abstract commands as logical procedures that guide the operation of each particular subsystem. These low-level sequences generate the control system output signals and read the input signals, supplying the Product System with logical responses that reflect the occurrence of uncontrollable events (Queiroz and Cury, 2002b).

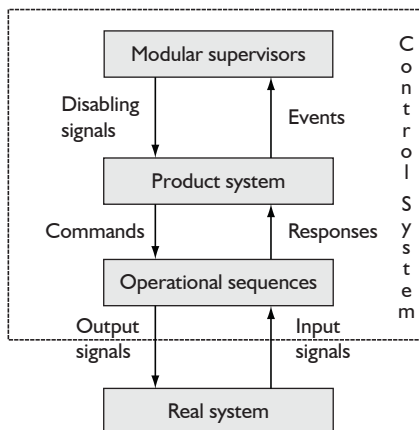


Figure 1 - Control structure proposed by Queiroz and Cury (2002b).

Proposed Development Approach

This section presents a development approach for control systems for reconfigurable processes in the context of agile manufacturing. In the first stage (modelling), the approach uses Formal Languages and an Automata Theory to represent the manufacturing system (for example, handling systems and processing cells) and the specifications (for example, operational and security requirements, and production routing). In this stage, models are modified when reconfigurations in processes are needed. In the second stage (synthesis), a specific approach (SCT and its extensions) is chosen according to the problem to be treated. Based on a specific approach, a single supervisor or local modular supervisors are obtained from manufacturing system and specification models, according to the SCT. In the third stage (implementation), occurs the translation of the theoretical supervisors into the own industrial platform's language (IEC 61131-3, 1998), according to a control structure shown in Figure 1. For validating purposes, the control structure is initially simulated, allowing the designer to substitute or modify the models. Gradually, parts of the structure are changed into control devices, which are attached to the real system. The implementation stage is finished in the moment that the control structure is fully accomplished.

Figure 2 shows the development cycle, which is characterized by three stages: modeling, synthesis and implementation. In the modeling stage we select from the subsystem and specification libraries a set of models to represent the real system and the application, respectively. In the synthesis stage, those models are used to generate local modular supervisors, according to the SCT (Ramadge and Wonham, 1989) and the Local Modular Control Theory (Queiroz and Cury, 2002a). In the implementation stage, the three levels of control structures (modular supervisors, product system and operational sequences) are integrated and gradually implemented in three steps: simulation, simulation and insertion of Control and Communication Technologies (CCT), and execution.

The control system development occurs cyclically in three stages – modeling, synthesis and implementation – up to the moment in which it complies with the real system's demanded application, thus resulting in an integrated and automated system. This development mode allows a continuous review of the results obtained in each step. By doing this, the designer can receive a new application (for example, a need for processes reconfiguration) and select new specification or subsystem models, which will appropriately comply with this new application.

A library of AMT gives support to the technological basis for the definition of operational sequences related to the subsystems, to the implementation of modular supervisors and to the needs for process reconfiguration. According to Gouvêa da Costa et al. (2000), the AMT concept includes numerical and computational based devices (software and hardware), designed to accomplish or support manufacturing activities and tasks. Communication

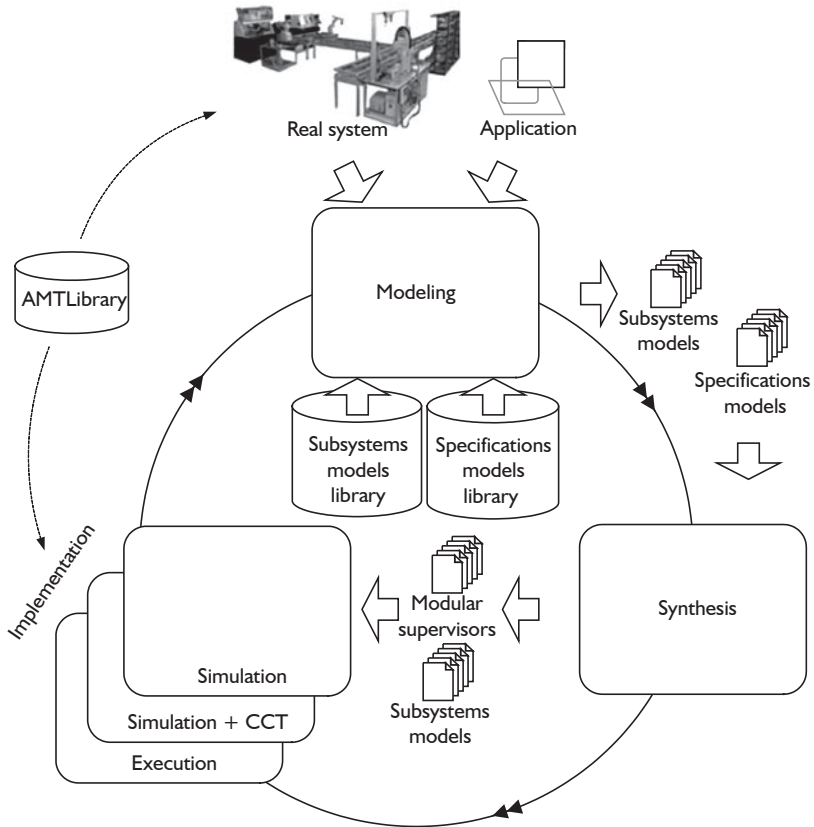


Figure 2 - Development cycle of control systems for reconfigurable processes.

technologies, programmable logical controllers (PLC), sensors and industrial actuators, are all examples of AMT.

Modeling stage

Automated manufacturing systems with characteristics of flexibility are made up of production cells, such as processing, welding, painting and assembly; of material handling systems, like conveyor belts, rotating tables, auto-guided vehicles (AGV) and automated storage and retrieval systems (AS/RS) (Groover, 2008) (Zhang et al., 2002). We can characterize a control system applied to manufacturing as being constituted by two levels: the supervision level and the subsystem level. The subsystem level is responsible for controlling the cells' local tasks and the material handling system. The supervision level is responsible for coordinating the cells' tasks, in such a way that the specifications (for instance, security and routing) are considered.

According to the modular approach, the several subsystems' open-loop behaviours (with no control) may be modelled with a set of asynchronous automata (with no common

event). This way, Queiroz and Cury (2002a) state that it is feasible to obtain a Product System Representation (PSR), according to what was discussed in the previous section. The steps that are needed for modeling are described as follows:

- 1) Identify the set of subsystems involved in the manufacturing system;
- 2) Build the Gi automaton of each subsystem i involved, as synthetically as possible;
- 3) Calculate the most refined Product System Representation (PSR) and produce the composition of synchronous subsystems; and
- 4) Model each isolated specification, considering only the relevant events.

According to the development cycle presented in Figure 2, we should initially select the adequate models related to all the subsystems that compose the real system. After that, we should select models that represent the specifications to be applied to the system, and these should define the way to coordinate all the subsystems. Therefore, the task of building appropriate models, which represent the subsystems, as well as each one of the specifications, is not a simple task and requires certain experience in modeling manufacturing systems (Cassandras and Lafortune, 2008). This way, despite the advantage of the SCT in the supervisor automatic synthesis, the construction of models for the real system and specifications might depend on the designer's experience and inspiration, thus compromising the reliability, necessary time and development global cost.

Modeling starts by identifying the subsystems that form the global plant. In many cases, this activity consists of a relatively simple task, once the real system's spatial configuration (for example, a manufacturing system composed by cells and stations) allows the designer to identify the several existing subsystems. The attention here shall be to consider the identification of each state of the subsystems. The coordination function the supervision system expects must be compatible with the correct identification of the subsystems' several states. In general, it is possible to identify the following subsystems' states:

- Inactive state (usually, the initial state);
- Active states (a subsystem may eventually have different functioning states); and
- Failure or break interruption state.

When modeling specifications, the construction of a set of models corresponds to the requirements to be imposed to the subsystems. Cassandras and Lafortune (2008) define the specification classes commonly present in a SED. The authors consider four cases in which the specification models are based: prohibited states, alternation of events, illegal chains and states refinement. In the first case, we identify states in the real system's model that cannot occur due to physical restrictions or security. The specification model is obtained by simply excluding those states from the system. In the second case, the coordination requirement imposes an alternation between events. For example, the need to alternate between two events α and β , when α occurs first, leads to the construction of a two-state automaton to accomplish this alternation. In the third case, we identify as

illegal all the chains in the real system model that contain certain sub-chains. In this last case, we need to memorize how a certain system state was reached so as to specify which future behaviours are acceptable. We should then refine such state in as many states as necessary.

In a previous work, Santos and Buseti (2004a,b) proposes the creation of libraries with specification models related to the manufacturing systems' physical configuration. The aspect that is mainly explored is the transportation mode used between these work cells. This way, the use of synchronous conveyors (Groover, 2008) leads to a particular group of specifications. The use of intermediate positions (due to physical restrictions or to a delay preview in the product transference) between work cells also generates particular specifications models. Santos and Buseti (2004a,b) explore other several possible configurations in manufacturing systems and for each one of these a specification model is associated. In addition to the use of libraries, the modeling step assumes that all subsystems that compose the manufacturing system are identified and have each one of them, an associated operational sequence.

Synthesis stage

The steps that are necessary for the synthesis are described as follows:

- 1) Obtain the local plant for each specification. To do that, it is necessary to compose the PSR subsystems that have events in common with a related specification;
- 2) For each local plant, calculate the language that fulfils the specification (by means of the synchronous product of each local plant and its respective specification);
- 3) Calculate the maximum controllable language contained in each local specification;
- 4) Check the resulting languages' local modularity;
- 5) If they are not modular, try to solve the problem using another approach; and
- 6) If they are modular, implement a local supervisor for each controllable language.

This stage consists of applying the synthesis procedure proposed by Ramadge and Wonham (1989) and Queiroz and Cury (2002a). Local modular supervisors are synthesized from the subsystem and specification models selected in the previous stage. Considering that each following step has the objective of implementing the supervisors in industrial platforms (for example, a PLC), it is still necessary to apply supervisor reduction algorithms (Vaz and Wonham, 1986; Su and Wonham, 2004), so as to obtain reduced supervisors (fewer states) with the same control action. This allows a smaller amount of memory and a better control program legibility.

Based on specific mathematical formalisms, we can still identify the tools related to the control structure synthesis. As an example, we have the control system synthesis process in an SCT, which can be accomplished through a TCT (Wonham, 1999), GRAIL (Raymond and Wood, 1996), DESCO (Fabian and Hellgreen, 2006), UKDES (Chandra et al., 2002), and

VER (Balemi et al., 1993). The computer tool SUPREMICA (Akeson, 2002) translates a formal language (automaton) into a normalized control language (IEC 61131-3, 1998).

Implementation stage

The control structure (supervisors and product system) obtained in the synthesis stage is initially implemented in an environment named Supervisor Code Generator (SupCoG). This consists of a computer platform that accomplishes two operations: it generates the control code from the TCT's tables that correspond to the supervisor and product systems; it sends and receives external signals that correspond to the commands and responses associated to the product system level. The implementation stage includes three steps, according to Figure 3.

In the implementation stage's first step, we simulate the control structure's three levels (modular supervisors, product system and operational sequences). To do that, the SupCoG is attached to a plant simulator (e.g. Arena[®], Pro Model[®], EM-Plant[®]) so that both may exchange signals. In the simulator, the subsystems are implemented, according to the real system's configuration, which was used for the product system generation. This way, the simulation occurs according to the restrictions imposed by the local modular supervisors. The simulation result allows us to evaluate the completion and correctness in the subsystem or specification models. The control structure simulation is useful either to do the first validation of the built models (subsystems and specifications) or to detect modifications and the necessary inclusions. The designer may start to run the local modular supervisor level and the product system, and follow the evolution of the states and associated control actions.

In the implementation stage's second step, the subsystems implemented in the plant simulator are progressively substituted by real components of the plant that runs the real operational sequences (e.g. PLC, PC-based control). This way, the SupCoG communicates simultaneously with the plant simulator and the real subsystems by inserting communication and control technologies. In this step, it is possible to progressively validate the control structure (software), when it is connected to the real system, and to analyse questions related to the distribution of the physical control system (hardware).

In the third step, the devices (e.g. PLC, PC-based control) that implement the operational sequences (OS) are completely attached to the respective sensors and actuators in the real subsystems. In this step, also occurs the translation of the modular supervisors set and the product system (a set of subsystems models) into their own programmable controllers' programming language (IEC 61131-3, 1998).

Methodology Experimental Application

The methodology that is proposed in this work is applied to the project and re-project of a manufacturing system prototype (reconfigurable processes). Initially, the project is

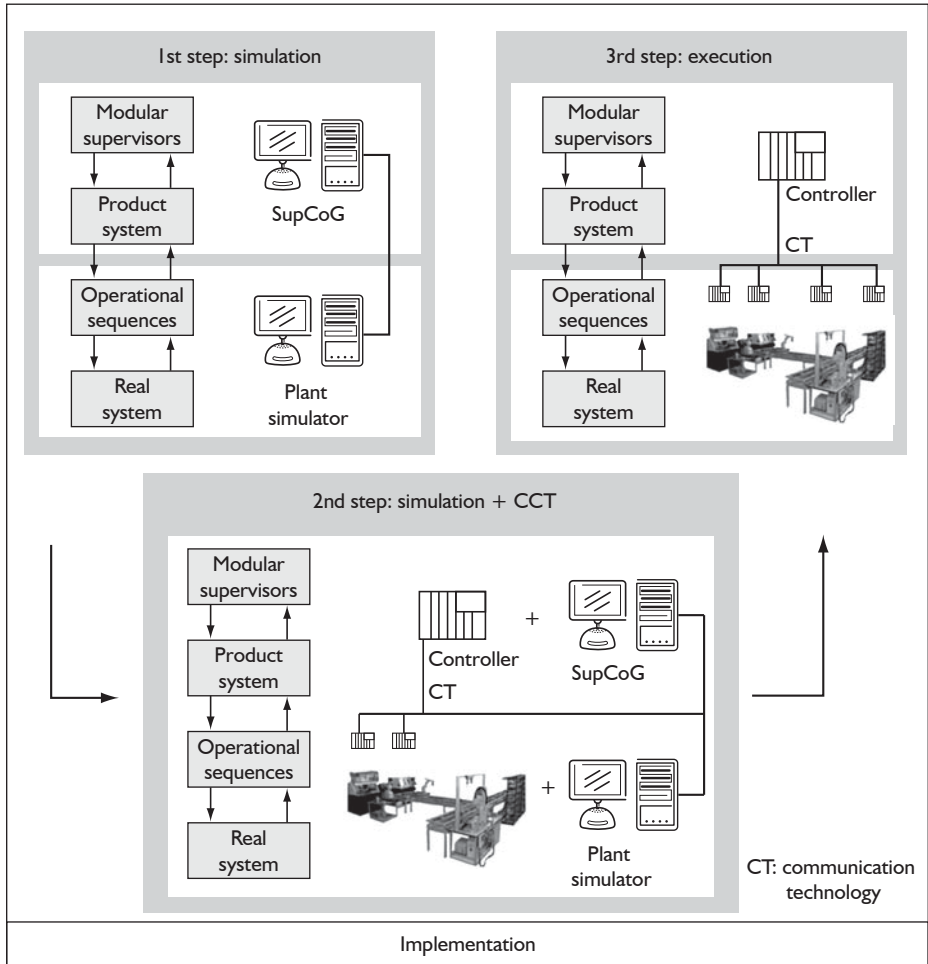


Figure 3 - Steps of implementation stage.

developed considering the production of a single product family (initial application). Next, due to a new market need (a new application), the re-project is developed considering the insertion of a new product family in the real system that already exists. This section presents the description of the real system (prototype), the applications demanded and the development cycle steps – modeling, synthesis and implementation – for the project and re-project of the reconfigurable processes.

Real system description (manufacturing system prototype)

The system prototype, which is presented in Figure 4, executes typical manufacturing operations: manufacturing processes, conveying, measurement, storage and classification. The system’s main purpose is to classify, to process and to store products according to certain attributes. The system is composed by six subsystems: material supply (G_1),

classification and measurement (G_2), conveying (G_3), processing 1 (G_4), processing 2 (G_5) and storage (G_6).

The G_1 subsystem has the purpose of storing and supplying raw materials with no classification to subsystem G_2 . Subsystem G_2 accomplishes two activities with the raw materials: it classifies them according to their type (colour and material) and measures their height. Measurement is done because these raw materials may not be uniform, and therefore, there might be different classes of dimensional tolerance. It is necessary to have a measurement device, once the G_2 subsystem may discard some materials that do not fit the desired tolerance. The classification of parts is done by combining sensor signals. For example, the metal part is identified through the reading of inductive, capacitive and optical sensors, while the red plastic part is read by the capacitive and optical sensors. After these operations are accomplished, a raw material goes to subsystem G_3 . Subsystem G_3 performs the conveying through subsystems G_2 , G_4 , G_5 and G_6 . Subsystems G_4 and G_5 accomplish specific manufacturing processes. Finally, subsystem G_6 stores the final product according to the attributes obtained in subsystem G_2 (measurement and classification) and to the accomplished production route.

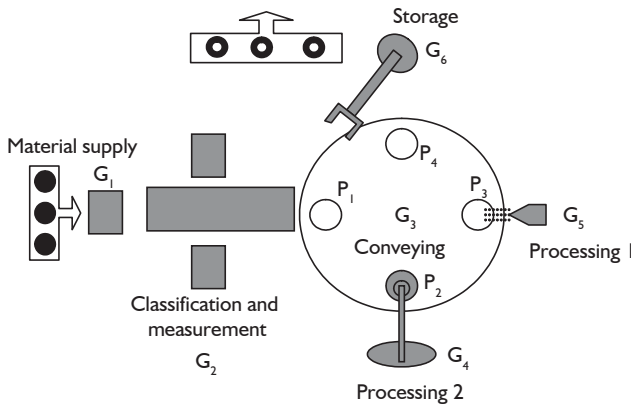


Figure 4 - Manufacturing system prototype.

Description of the demanded applications

There are three types of raw material: the MPA (metal), MPB (black plastic) and MPC (red plastic) materials. We initially consider that these raw materials are stored in a single magazine without classification. The initial application needs the production of three types of final product: PFA1, PFB1 and PFC1, which are related to the three raw materials MPA, MPB and MPC, respectively, and also, to production route R1. Due to a market demand, the new application needs the production of three other types of final product: PFA2, PFB2 and PFC2, related to the three raw materials MPA, MPB and MPC, respectively, and also to production route R2. According to the external demand, the user defines the type and quantity of the final product to be produced. Table 1 presents a summary of the final products.

Table I - Description of raw materials and the associated production routes.

Final product	Raw material	Production route
PFA1	MPA	R1
PFA2	MPA	R2
PFB1	MPB	R1
PFB2	MPB	R2
PFC1	MPC	R1
PFC2	MPC	R2

Production route R1 uses the G_4 and G_5 subsystems' manufacturing processes, and R2 uses only the G_4 subsystem's manufacturing process. Figure 5 illustrates the manufacturing system prototype with production routes R1 and R2.

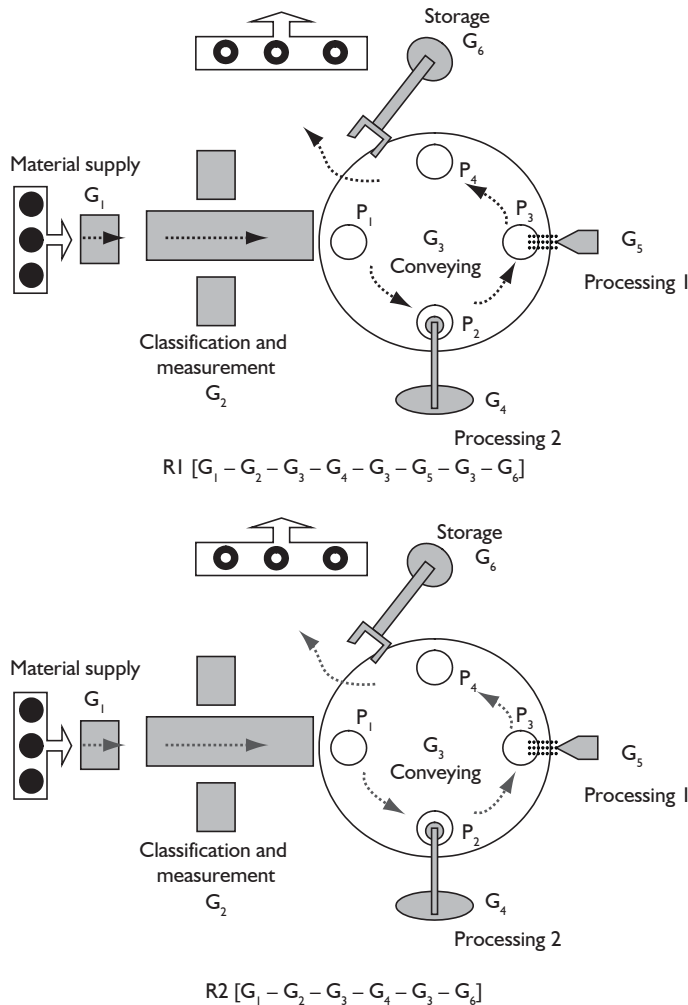


Figure 5 - Production routes.

Modeling stage

According to the proposed methodology, the first step (modeling) consists of representing, in automata, the subsystems that compose the real system and the specifications set (application). For this, we use the model library proposed in the present work. For the subsystems, the two-state model (active and inactive states) may be selected. They are presented in Figure 6.

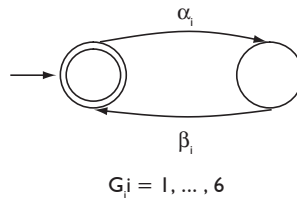


Figure 6 - Subsystem model.

In order to model the specifications, we analyse the system’s physical configuration, the production route and the coordination restrictions, in such a way that the material flow through the subsystems is correctly represented. In the initial application, for the correct material flow between subsystems G_1 and G_2 , a mutual exclusion specification may be selected. This specification imposes that G_1 and G_2 cannot work simultaneously and, at the same time, it defines the execution sequence for the activities related to these subsystems. After subsystem G_2 has accomplished the measurement and classification steps, the raw material goes to the conveying subsystem G_3 . This subsystem is defined as a synchronous conveyor (rotating table) (Groover, 2008). Such subsystem, moves a product from one position to another, and moves the other products to the subsequent positions. At this moment, the designer selects models considering the following information: four-position synchronous conveyor; first position for parts arrives, second and third positions for the processes and fourth position for removing parts. The specifications set, related to the synchronous conveyor, determines the correct material flow between subsystems G_2 , G_4 , G_5 and G_6 . The models for the nine necessary specifications ($E_a, E_b, E_{c1}, E_{c2}, E_{c3}, E_{c4}, E_{d1}, E_{d2}, E_{d3}$) for the initial application are presented in Figure 7.

In the new application, we keep the mutual exclusion specification between subsystems G_1 and G_2 . After subsystem G_2 has accomplished the classification and measurement steps, the raw material goes to the conveying subsystem G_3 . In this new application, the designer selects models considering the following information: a four-position synchronous conveyor; first position where the pieces arrive, second position for processing, third position with no operation being executed and fourth position for removing parts. The specifications set, related to the synchronous conveyor, determines the correct material

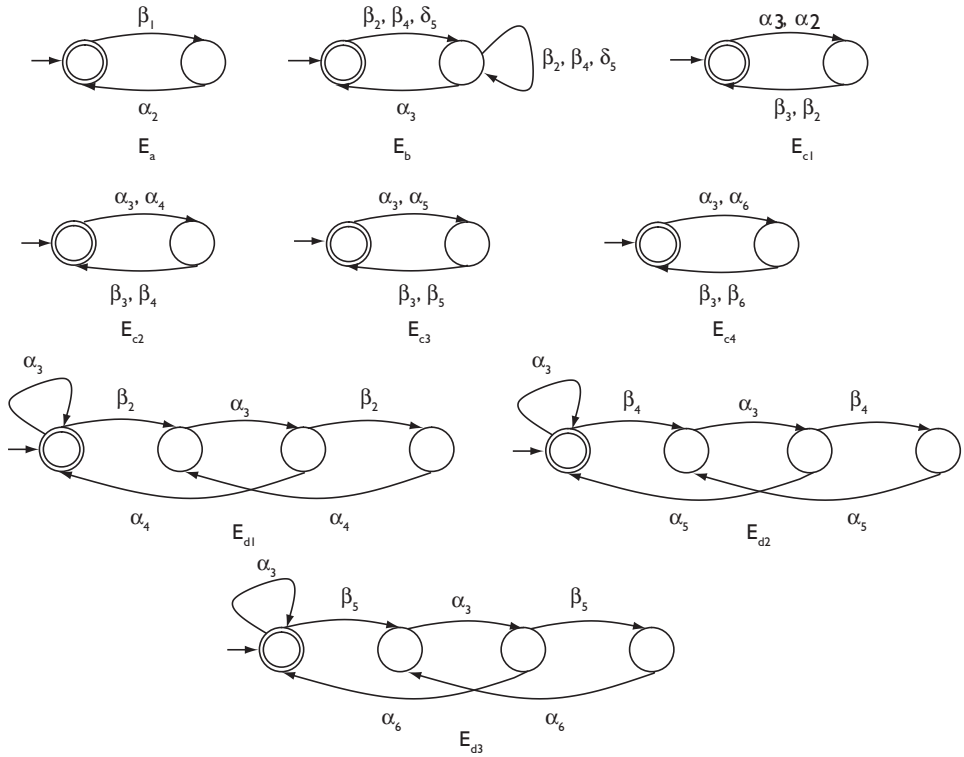


Figure 7 - Specifications models for the initial application.

flow between subsystems G_2 , G_4 and G_6 . The models for the nine necessary specifications ($E_a, E_b, E_{c1}, E_{c2}, E_{c3}, E_{c4}, E_{d1}, E_{d2}, E_{d3}$) for the new application are presented in Figure 8.

Synthesis stage

The next step – synthesis – corresponds to the SCT application and the local modular approach. This way, for each selected specification (initial application and new application), a supervisor is obtained by using the TCT tool. The joined action between the supervisors that were obtained restricted the subsystems behaviour (which composes the real system) to the respective applications.

For the initial application, nine supervisors are synthesized, each one of them with a specific purpose, from the selected specifications, which are presented in Figure 7. For example, considering specification E_{d1} , the local plant is obtained through the synchronous composition of the corresponding automata for the subsystems that share events with this specification ($G_{loc,d1} = G_2 \parallel G_3 \parallel G_4$). The local specification is obtained through the generic specification composition E_{di} with the corresponding local plant ($E_{loc,d1} = E_{d1} \parallel G_{loc,d1}$). One may then calculate the maximum controllable language contained in the specification, which is $SupC(E_{loc,d1}, G_{loc,d1})$. Next, through a supervisor minimization algorithm (Vaz and

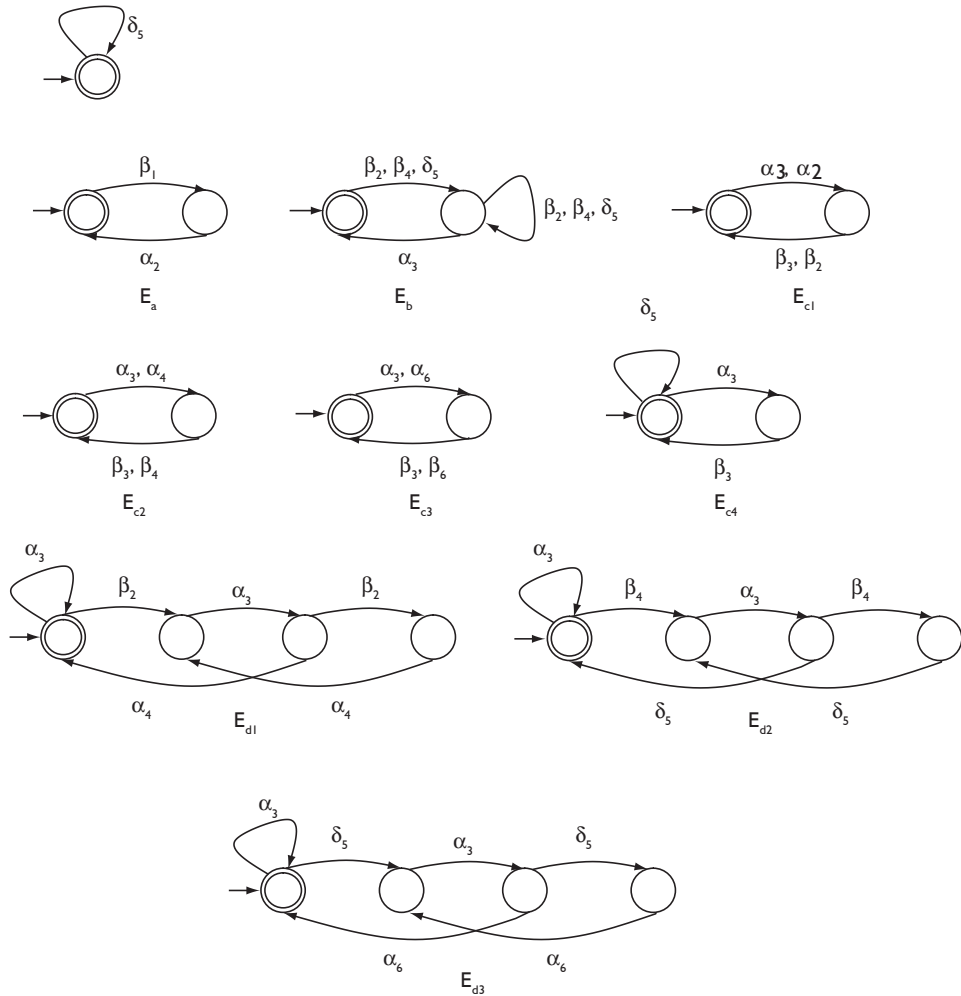


Figure 8 - Specification models for the new application.

Table 2 - Models used in the synthesis procedure and local supervisors obtained (initial application).

Generic specification	Local plant	Local specification	Supremal controllable language (local supervisor)
E_a	$G_{loc,a} = G_1 \parallel G_2$	$E_{loc,a} = G_{loc,a} \parallel E_a$	$SupC(G_{loc,a}, E_{loc,a})$
E_b	$G_{loc,b} = G_2 \parallel G_3 \parallel G_4 \parallel G_5$	$E_{loc,b} = G_{loc,b} \parallel E_b$	$SupC(G_{loc,b}, E_{loc,b})$
E_{c1}	$G_{loc,c1} = G_2 \parallel G_3$	$E_{loc,c1} = G_{loc,c1} \parallel E_{c1}$	$SupC(G_{loc,c1}, E_{loc,c1})$
E_{c2}	$G_{loc,c2} = G_3 \parallel G_4$	$E_{loc,c2} = G_{loc,c2} \parallel E_{c2}$	$SupC(G_{loc,c2}, E_{loc,c2})$
E_{c3}	$G_{loc,c3} = G_3 \parallel G_5$	$E_{loc,c3} = G_{loc,c3} \parallel E_{c3}$	$SupC(G_{loc,c3}, E_{loc,c3})$
E_{c4}	$G_{loc,c4} = G_3 \parallel G_6$	$E_{loc,c4} = G_{loc,c4} \parallel E_{c4}$	$SupC(G_{loc,c4}, E_{loc,c4})$
E_{d1}	$G_{loc,d1} = G_2 \parallel G_3 \parallel G_4$	$E_{loc,d1} = G_{loc,d1} \parallel E_{d1}$	$SupC(G_{loc,d1}, E_{loc,d1})$
E_{d2}	$G_{loc,d2} = G_3 \parallel G_4 \parallel G_5$	$E_{loc,d2} = G_{loc,d2} \parallel E_{d2}$	$SupC(G_{loc,d2}, E_{loc,d2})$
E_{d3}	$G_{loc,d3} = G_3 \parallel G_5 \parallel G_6$	$E_{loc,d3} = G_{loc,d3} \parallel E_{d3}$	$SupC(G_{loc,d3}, E_{loc,d3})$

Wonham, 1986), we obtained a supervisor with a smaller number of states and with the same controlling action. Table 2 presents the supervisors calculated from the subsystems models and the selected specifications for the initial application.

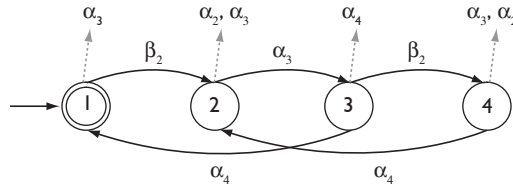


Figure 9 - Supervisor $\text{SupC}(G_{\text{loc,c1}}, E_{\text{loc,c1}})$ for the initial application.

As an example, Figure 9 presents the $\text{SupC}(E_{\text{loc,d1}}, G_{\text{loc,d1}})$ supervisor, which is the result of specification E_{d1} . The dashed line indicates the supervisor controlling action, which is to disable controllable events from subsystems G_2 , G_3 and G_4 .

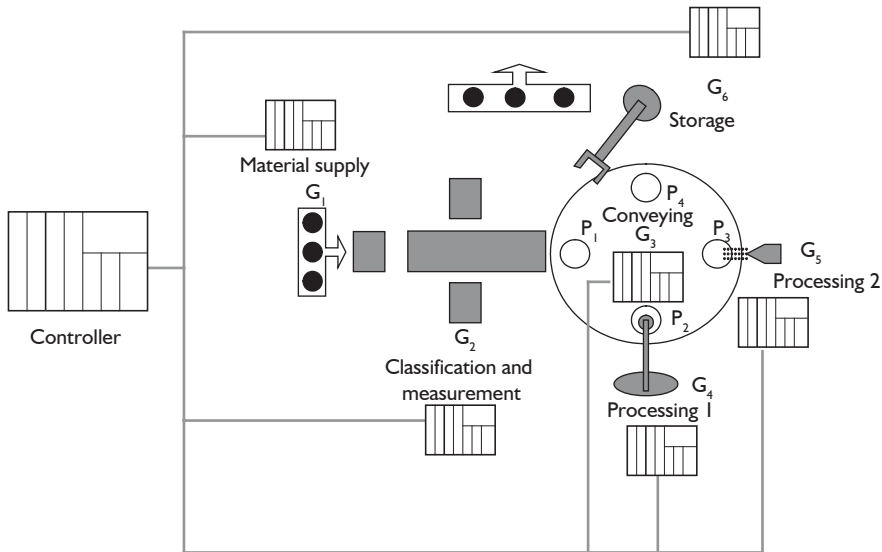
For the new application, nine supervisors are also synthesized, each one with a specific purpose, from the selected specifications presented in Figure 8. The same way as in the initial application, nine modular supervisors for the new application are synthesized.

Implementation stage

According to the proposed methodology, in the first step we simulate the control structure's three levels (modular supervisors, product system and operational sequences). To do that, the SupCoG is attached to EM-Plant simulator. The subsystems G_1, G_2, G_3, G_4, G_5 and G_6 are implemented in the simulator and the SupCoG runs the nine modular supervisors that accomplish the initial application. The simulation result shows us the correctness of the subsystem and specification models.

In the second step, each subsystem implemented in the EM-plant is progressively substituted by a PLC (Allen Bradley Micrologix 1200) (Micrologix, 2006) that runs the related operational sequence. This way, the SupCoG communicates simultaneously with the EM-plant and the PLC by means industrial Ethernet (Lee and Lee, 2002). In the third step, the six subsystems ($G_i, i = 1, \dots, 6$) implemented in the EM-plant are completely substituted by six PLC. Also, the control structure (modular supervisors and product systems) that runs in the SupCoG is completely executed in a controller (PLC), as Figure 10 illustrates.

When the new application is demanded, the three steps are performed again. In this case, the cycle development time is decreased because the subsystem models and a specification subset used in the initial application are reutilized. Also, the physical structure of the new and the initial applications are the same, according to the Figure 10. However, the PLC that runs the operational sequence related to subsystem G_5 (processing 2) is not used in the new application.



CT: Communication technology

Figure 10 - Physical control structure after the implementation stage.

Conclusions

This work presented a project methodology applied to reconfigurable processes in agile manufacturing systems. The insertion of the Supervisory Control Theory (SCT), computational tools for synthesis and simulation, subsystems and specifications models libraries, brought a reduction of the manufacturing system's development time. The development cycle is accomplished by integrating mathematical formalisms (Automata, Formal Languages and SCT), computational tools (synthesis, simulation), automata libraries (models of subsystems and specifications), Advanced Manufacturing Technologies (AMT) (Gouvêa da Costa et al., 2000) and simulation techniques.

The SCT is a formal approach that allows automatic synthesis of supervisors. Allied to the SCT, the modular approach brings a greater agility to the project, an advantage for the local modular approach. Model libraries make the modeling step easier and allow models to be reused in subsequent projects. The integration of the simulation and implementation steps allowed a greater liability for validating, optimizing and accomplishing the control structure.

The experiment performed shows that the objective of this paper has been reached. After executing a set of control rules to manufacture a product, a demanding for a new product was stated. Following the proposed methodology, the manufacturing system has been changed (reconfigurable processes) and the new demand could be accomplished. The subsystems and specifications models used will be able to model some other manufacturing

systems in subsequent projects. Also, it is possible to use the same models (or part of them) in the same manufacturing system whenever a new demand is established.

It is important to remark that a set of hardware and software has been built to support the proposed methodology. According to Figure 3, it was specified a a codification and communication between SupCoG and the plant simulator (1st step – simulation), a communication among computational and hardware platforms (2nd and 3rd steps). In fact, a new environment of hardware and software has been established in the context of the proposed methodology, in order to perform a reconfiguration of integrated and automated systems.

The proposed methodology still presents some limitations and this allows us to presume the continuity of the work that was developed. In the beginning of the cycle, we shall study more deeply the criteria for segmenting the real system into subsystems, as well as the identification of a specification set for a certain application. Subsystems and specifications model libraries shall be constantly updated. The experience with each new project shall be used to update those libraries. Therefore, the selection of models still depends very much on the designer's experience. We shall research some methods to systematize the choice of model libraries from the real system and from the application. The library of Advanced Manufacturing Technologies (AMT) shall also be constantly reviewed and updated. A systematic approach shall be researched to define the communication and control technologies needed to implement and accomplish the modular control structure.

As a result of this methodological approach, we have the necessary elements for the documentation of the technical project. For such purpose, there are still other factors to be considered, such as maintenance, commercial relations, costs, updates and technical reviews of the equipment already installed.

References

- Akesson, K. (2002) *Methods and Tools in Supervisory Control Theory*. PhD Thesis, Chalmers University of Technology, Sweden.
- Balemi, S. et al. (1993) Supervisory control of a rapid thermal multiprocessor. *IEEE Transactions on Automatic Control*, Vol. 38, No. 7, pp. 1040-1059.
- Bi, Z.M. et al. (2007) Reconfigurable Manufacturing Systems: The State of the Art. *International Journal of Production Research*, Vol. 46, No. 4, pp. 967-992.
- Carrol, J. and Long, D. (1989) *Theory of Finite Automata*. Prentice-Hall International Editions, London.
- Cassandras, C. G. and Lafortune, S. (2008) *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, Boston.
- Chandra, V. et al. (2002) A graphical software tool for the design, analysis and control of discrete event systems. Available: <http://clue.eng.iastate.edu/~rkumar/PUBS>. Access: 2002.

- Chun-Che, H. and Kusiak, A. (1998) Modularity in Design of Products and Systems. IEEE Transactions on Systems, Man, and Cybernetics – Part A, Vol. 28, No. 1, pp. 912-931.
- Erbe, H. H. (2002) Low Cost Intelligent Automation in Manufacturing. Proceedings of the 15th Triennial IFAC World Congress, Barcelona, Spain.
- Fabian, M. and Hellgren, A. (2006) Desco – a Tool for Education and Control of Discrete Event Systems. Kluwer Academic Publishers, Boston.
- Gamberi, M. et al. (2008) A new approach to the automatic analysis and control of material handling systems: Integrated Layout Flow Analysis (ILFA). International Journal Advanced Manufacturing Technology.
- Goyon, D. et al. (2004) Pragmatic approach for modular control synthesis and implementation. International Journal of Production Research, Vol. 42, No. 145, pp. 194-203.
- Gouvêa da Costa, S. et al. (2000) Advanced Manufacturing Technology: defining the object and positioning it as an element of manufacturing strategy. VI International Conference on Industrial Engineering and Operations Management – VI ICIEOM. ABEPRO, São Paulo.
- Groover, M. (2008) Automation, Production Systems, and Computer-Integrated Manufacturing. Prentice-Hall, New Jersey.
- Hoda, A. and El Maraghy (2006) Flexible and reconfigurable manufacturing systems paradigms. International Journal Flexible Manufacturing Systems, Vol. 17, No.2, pp. 261-276.
- Holloway, L.E. et al. (1997) A survey of Petri nets methods for controlled discrete event systems. Discrete Event Dynamic Systems: Theory and Applications. Kluwer Academic Publishers, Boston. pp. 151-190.
- IEC61131-3 International Electrotechnical Commission (1998) Programmable Controllers - Programming Languages. London.
- Kelkar, A. et al. (2005) Geometric Algorithms for Rapidly Reconfigurable Mold Manufacturing of Free-form Objects. Computer Aided Design, Vol. 37, No. 1, pp. 211-234.
- Kusiak, A. and Chun-Che, H. (1996) Development of Modular Products. IEEE Transactions on Components, Packing, and manufacturing Technology – Part A, Vol. 19, No. 4, pp. 833-848.
- Kusiak, A. and Salustri, F.A. (2007) Computational Intelligence in Product Design Engineering: Review and Trends. IEEE Transactions on Systems, Man, and Cybernetics: Part C, Vol. 37, No. 5, pp. 766 -778.
- Lee, K.C. and Lee, S. (2002) Performance evaluation of switched Ethernet for real-time industrial communications. Computer Standards and Interfaces, No. 24, pp. 411-423.
- Manzini, R. et al. (2004) Framework for designing a flexible cellular assembly system. International Journal of Production Research, Vol. 42, No. 17, pp. 303-331.

- Manzini, R. et al. (2005) Design and control of a flexible order picking system (FOPS): a new integrated approach to the implementation of an expert system. *Integrated Manufacturing Systems*, Vol. 16, No. 12, pp. 18-35.
- Allen-Bradley (2006) Micrologix 1200 and Micrologix 1500 Programmable Controllers. Instruction Set Reference Manual. Sacramento. Available: <http://www.ab.com/micrologix>. Access: 2006.
- Minhas, R.S. (2002) Complexity Reduction in Discrete Event Systems. PhD. Thesis, Dep. of Electrical and Computer Engineering, University of Toronto, Canada.
- Moore, P.R. et al. (2003) Virtual engineering: an integrated approach to agile manufacturing machinery design and control. *Mechatronics*, Vol. 13, No. 9, pp. 1105-1121.
- Ollero, A. et al. (2006) Mechatronics, robotics and components for automation and control. IFAC milestone report, *Annual Reviews in Control*, Vol. 30, No. 1, pp. 41-54.
- Siemens (2005) Profibus. Available: <http://www.profibus.org/>. Access: July 1, 2005.
- Queiroz, M. and Cury, J. (2002a) Controle Supervisório Modular de Sistemas de Manufatura. *Revista Controle and Automação, SBA*, Vol. 13, No. 1, pp. 115-125.
- Queiroz, M. and Cury, J. (2000) Modular supervisory control of large scale discrete-event systems. *IFAC 5th International Workshop on Discrete Event Systems: Analysis and Control*. Kluwer Academic Publishers, London, pp. 103-110.
- Queiroz, M. and Cury, J. (2002b) Synthesis and implementation of local modular supervisory control for a manufacturing cell. *Discrete Event Systems: Analysis and Control*. Kluwer Academic Publishers, Boston, pp. 103-110.
- Ramadge, P.J. and Wonham, W.M. (1989) The control of discrete event systems. *Proceedings of IEEE*, Vol. 17, Special Issue on Discrete Event Dynamic Systems, pp. 81-98.
- Rao, Y. et al. (2006) Agile manufacturing system control based on cell re-configuration. *International Journal of Production Research*, Vol. 44, No. 10, pp. 1881-1905.
- Raymond, D. and Wood, D. (1996) Grail: Engineering automata in C++: Version 2.5. Available: <http://www.csd.uwo.ca/research/grail>. Access: January 31, 2003.
- Sanchez, L.M. and Nagi R. (2001) A review of agile manufacturing systems. *International Journal of Production Research*, Vol. 39, No. 9, pp. 1567-1585.
- Santos, E.A.P. and Buseti, M.A. (2004a) Modeling the Control of a Discrete Event Manufacturing Cell. *Conference on Management and Control of Production and Logistics*, Santiago.
- Santos, E.A.P. and Buseti, M.A. (2004b) An integrated design methodology for automated manufacturing systems. *Conference on Management and Control of Production and Logistics*, Santiago.
- Su, R. and Wonham, W.M. Supervisor Reduction for Discrete-Event Systems. *Discrete Event Dynamic Systems*, Vol. 14, No. 1, pp. 31-53.
- Vaz, A.F. and Wonham, W.M. (1986) On supervisor reduction in discrete-event systems. *International Journal of Control*, Vol. 44, No. 2, pp. 475-491.

- Wadhawa, S. et al. (2005) Flexibility-enabled lead-time reduction in flexible systems. *International Journal of Production Research*, Vol. 43, No. 15, pp. 3131-3162.
- Wang, A. et al. (2005a) Complex Assembly Variant Design in Agile Manufacturing. Part I: System Architecture and Assembly Modeling Methodology. *IIE Transactions on Design and Manufacturing*, Vol. 37, No. 1, pp. 1-16.
- Wang, A. et al. (2005b) Complex Assembly Variant Design in Agile Manufacturing. Part II: Assembly Variant Design Methodology. *IIE Transactions on Design and Manufacturing*, Vol. 37, No. 1, pp. 17-34.
- Wonham, W.M. (1999) Notes on control of discrete event systems. Course notes for ECE 1636F/1637S, Dept. of Electrical and Computer Engineering, University of Toronto, Canada.
- Zhang, J. et al. (2002) Investigation of the reconfigurable control system for an agile manufacturing cell. *International Journal of Production Research*, Vol. 40, No. 3, pp. 3709-3723.

Biography

Marco Antonio Buseti de Paula obtained his Dr.-Ing in electrical engineering (mechatronics) in the Universität Gesamthochschule Paderborn (1994). Currently, he is a Professor in the post-graduate program in production engineering and systems in the *Pontifícia Universidade Católica do Paraná* as well as Adjunct Professor at *Universidade Tecnológica Federal do Paraná*. His areas of interest are integration, automation, and system evaluation.

Eduardo Alves Portela Santos is a doutor in electrical engineering by the *Universidade Federal de Santa Catarina* (2003). Currently, he is a Professor in the post-graduate program in production engineering and systems in the *Pontifícia Universidade Católica do Paraná* as well as Adjunct Professor at *Universidade Tecnológica Federal do Paraná*. His areas of interest are integration, automation, and system evaluation.

Submitted: 7th August, 2006.

Accepted: 27th April, 2009.