Batch Process Modelling Using Petri Nets.

D. ANDREU *, J.C. PASCAL *, H. PINGAUD **, R. VALETTE *
* LAAS-CNRS, 7 av du Colonel Roche 31077 Toulouse Cedex, France
** LEAP, ENSIGC, chemin de la loge 31078 Toulouse Cedex, France
(e.mail : andreu @ laas.fr)

Abstract: The automation of batch processes poses difficult issues mainly because it is necessary to concurrently operate with continuous and discrete models. In this paper, we analyse how the hierarchical approach used in the field of discrete manufacturing systems can be extended to batch systems. This hierarchical approach gave rise to three levels which are composed by a discrete event model modelling the control policy (discrete view of the process) and by a continuous model of the process (continuous view of the process). The consistency between the continuous models of the process and the physical state of the plant is ensured by an event generator. This event generator is presented and various cases of variables are considered. Finally, a discrete event modelling of the control policies is illustrated through an example of a batch multipurpose plant, using Petri Nets.

Keywords: Batch process, hierarchical structure, control, model of the process, Petri nets, event generator.

1. Introduction

Batch processes are currently used in the chemical and food process industries (CPI). Their automation and optimization pose difficult issues mainly because it is necessary to operate concurrently with continuous (algebraic or differential equations) and discrete (state machines) models.

The aim of this paper is to analyse how techniques developed in the field of discrete manufacturing systems (DMS) can be extended to batch systems. Emphasis will be put on the possibility of using the typical hierarchical Petri Net based approach for describing the discrete aspect of batch systems from local to supervisory control.Another major issue is the consistency between continuous and discrete models evolution.

2. Batch processes versus DMS

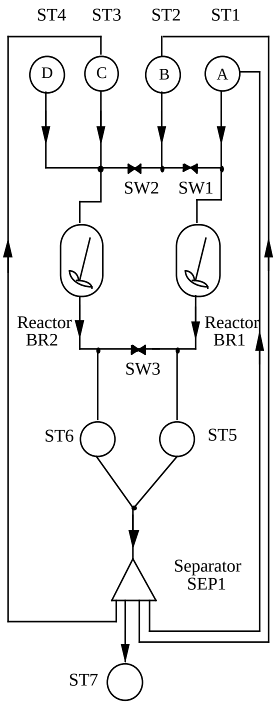
The first characteristic of batch processes in CPI is that the batches are rarely a numbered collection of entities (parts) but mainly amounts of stuff (liquid, gas,...)[1]. Batches are defined as threshold in unit devices (storage units, reactors,...). During each transformation (in a reactor, in a separator,...), the amount of material may vary because products (solvent) are added or separated, etc... In discrete manufacturing systems, a batch denoted as a token at a higher level of abstraction can be split down into n tokens (the n items) at a lower level. This is not possible in batch systems.

Another major difference is the physical layout importance. In DMS, during the transport operation, the concerned unit devices are independent to each other, the transport device (conveyor, guided vehicle,...) containing the product moves freely from a device to another one. It is not the case in batch systems where the transfer device is fixed and the stuff passes through it. In consequence the allocation of unit devices to operations are not independent to each other: unit devices are synchronised and linked during the whole transfer operation duration. As a consequence, the flowsheet and the switching valves system are of major importance whereas the physical layout of the machines are rarely taken into account in discrete control.

However at a sufficiently high level of abstraction, a batch system can be seen as series of discrete operations delimited by events (open a valve, begin heating, etc...) in the same way as discrete event systems (DES).

3. Example

The presented approach and concepts will be illustrated through the following example of a batch production on a multipurpose plant. This example has been presented in detail in [2]. The flowsheet is shown in figure 1.



- Figure 1: Flowsheet -

Raw materials A, B and D are stored in tanks ST1, ST2 and ST4 respectively. In the beginning both reactors, BR1 and BR2, carry out the fabrication of intermediate C: A+B->C. The reaction mixture containing unreacted A and B, and the intermediate C is stored in ST5.

After the level in the tank has reached $45m^3$, the separator SEP1 is activated (separation of C). SEP1 splits the input stream into three streams, one rich in A, the other rich in B and the third pure C. The intermediate C is stored in ST3. The concentration of A in ST1 and of B in ST2 change continuously because the recycle stream concentrations change over time. After the amount of C has reached $35m^3$, the reactor BR2 has to exclusively carry out the manufacture of product E: C+D->E. This product is stored in ST6, and is accumulated until the level reaches 20 m^3. Then the separator separates product E into pure C, which is sent back to ST3, and a mixture of D and E, which is sent to ST7. Both separations compete for separator SEP1.

The recipe for the batch reactors is as follows: a) load raw materials, b) preheat, c) react, d) cool, e) store the product, d) clean the reactor.

4. The hierarchy

Due to the complexity of the CPI control, a hierarchical approach allows to decompose the whole problem into smaller subproblems according to different competence levels. Moreover it reduces the information transfer between the decision centers.

The discontinuous processing is constituted by elementary operations of continuous nature. They have a finite duration, and their location is limited to a subset of apparatus of the plant, as for example the transformation in a batch reactor. This discontinuity, and the significant amount of events and actions of discrete nature allow to consider differents DES approaches on Continuous-Discrete Systems (CDS). The analogy with the DES hierarchical control gave rise to three levels.

Each level is composed by a discrete event model modelling the control policy and giving a discrete view of the process, and by a continuous model giving a continuous view of the process. This continuous model of the process is used to ensure the consistency between the control policies and the actual physical state of the plant. Its accuracy degree depends on the considered level of abstraction: from an overall view at the upper level to a detailed view at the lower one.

Let us note that each level contains a process behavior monitoring function which will not be detailled in this paper.

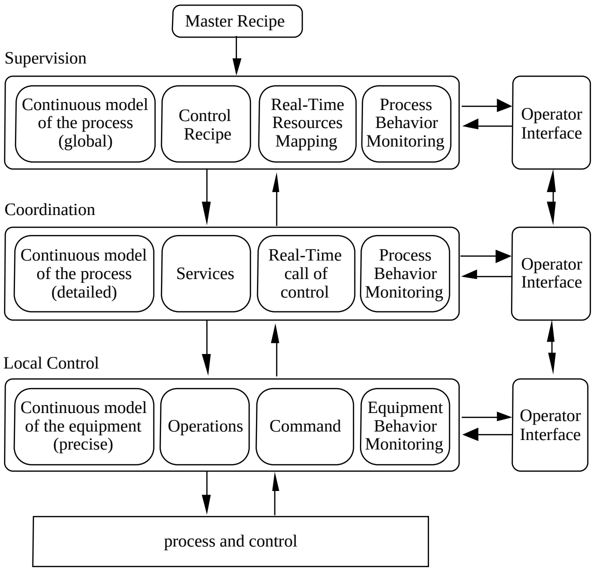
The lowest level (level 1) models the device operation: this level will be called the local control level. It contains the elementary operations, continuous ones (local regulation functions) or discrete ones (open valve,...). The level 0 corresponds to the I/O subsystem.

The level 2, also called the coordination level, contains the services. The phases describe the sequences of macro-operations which are encapsulating the elementary operations in order to build basic functions. These basic functions are seen as services by the upper level, as for example the reactor feeding function. A complete services library of all the possible operations on the devices has to be defined, allowing to know what kind of operation can be carried out by each entity (such as reactors, centrifuges, etc). This services library is developed for each unit of the plant.

The level 3 which will be called the supervision level contains the ’control’ recipes and functions which ensure a consistent working of the plant. An example of such a function is the real-time mapping of resources (unit allocations, switching valves and manifold management,...) to the operations.

These ’control’ recipes are combinations of services in order to implement ’master’ recipes.

Several types of recipes can be defined in batch processing [3]. We will use two types of recipes: the ’master’ recipe and the ’control’ one. The ’master’ recipe contains the procedure to be followed, i.e. it represents the phase sequences. The ’master’ recipe assigns virtually the equipments and indicates the various ingredients which have to be used to make the product. It also contains product related data such as the parameters of the product transformation (formulas, times, temperatures, quantities,...). The 'control' recipe corresponds to the real - time execution of the previous one with an effective equipment assignement.


Master Recipe
Supervision
Continuous model of the process (global)
Control Recipe
Real-Time Resources Mapping
Process Behavior Monitoring
Operator Interface
Coordination
Continuous model of the process (detailed)
Services
Real-Time call of control
Process Behavior Monitoring
Operator Interface
Local Control
Continuous model of the equipment (precise)
Operations
Command
Equipment Behavior Monitoring
Operator Interface
process and control

- Figure 2 - hierarchical structure-

5. Some issues about the hierarchy.

The interface between "discrete" and "continuous" views differs in accordance with the modelling aim, i.e. simulation or real-time control. The complexity comes from the mutual interaction of events resulting from the continuous evolution (thresholds, time out, etc...) and others which are a consequence of the discrete model evolution (control decisions).

In case of simulation, the models depend on the nature of the simulated activity (continuous or discrete) and on the considered level of abstraction [4].

Let us analyse the discrete-continuous interface in case of real-time control.

5.1 *The discrete-continuous interface in real-time control*

Let us consider the example in ¤3, whose flowsheet is shown on figure 1.

A storage tank ST5 contains a product C obtained by the reaction of two raw materials A and B. The obtained product C is not pure due to a proportion of A and B that have not reacted. It is necessary to separate them in order to isolate pure C. Then, the obtained pure C is stored in another tank ST3 whose level is measured by a continuous sensor. When the level of pure C in ST3 reaches a particular threshold, it is possible to start the fabrication of another product E.

At the end of the separation of the product C, a report is returned. This report allows to consider that a batch has been separated but it is not possible to deduce a priori that the separation has indeed produced the wished volume in ST3. The information which is available at the upper level of the hierarchy is not sufficient to make a decision.

Thus, in order to make decisions or to make choices, each level has to reconstitute a consistent information about the system, and/or the product state. The hierarchy may be by-passed. For instance, the supervisory level has to access the value of the storage tank sensor to ensure the consistency between the production strategy and the actual physical state of the plant.

The sensor state value has different meanings at the lower level and at the upper one: in the first case it represents a continuous data (sampled) used for continuous regulation, in the second case it defines the value of a predicate used in a decision-making policy.

So we notice that the continuous-discrete interface is not located at a particular level, and this partially contradicts the hierarchical approach.

As seen in ¤4, the consistency between the production strategy (or the discrete event control policy) and the actual physical state of the plant is ensured by means of the continuous models of the process at each hierarchical level.

As a consequence, the consistency between the actual physical state of the process and its continuous model is the major issue.

From a distributed implementation point of view the supervisory control cannot directly access the sensors. Indeed, the continuous-discrete conversion is classically located and necessarily done at the low level. So, a tool is required to create a new type of relation between the process on one hand and its control part on the other.

Consequently, the interface between the process and its continuous models has to be represented by an event generator (EG). This EG is based on the evolution of continuous variables which characterise the system state, and generates events when particular thresholds are reached for example.

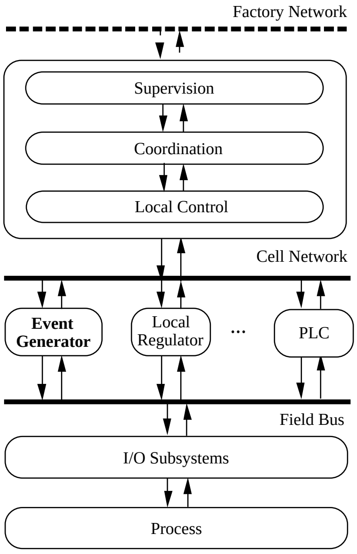
5.2 *The event generator*

5.2.1 *The context description*

Distributed computer control systems are generally based on an hierarchy of Local Area Networks. In order to ensure the temporal and spatial consistency of the real-time data base it is better to avoid the simultaneous handling of periodic and aperiodic messages. The best choice is to have a Time Triggered (TT) policy for the field bus and an Event Triggered (ET) policy for the cell network [5].

A TT policy observes the state of the environment periodically (depending on the nature of the critical timing constraints imposed by the external environment) and initiates system activities at reccuring predetermined points in the globally synchronised time. The deadlines dictated by the dynamics of the controlled entity (its environment) must be guaranteed [6].

In an ET policy a system activity is initiated as an immediate consequence of the occurence of a significant event in a Real-Time Entity (RTE) [6]. An event can come from the RTE environment or from its computer systems (command report, occurence of any detected abnormality,...).



- Figure 3 : physical control -

5.2.2 *The significant events*

A significant event is a change of state in a RT entity which has to be handled by the computer system. Other state changes of the RT entity are considered insignificant and are neglected [7]. As a consequence, the significant events ensuring the consistency constraint between the evolution of continuous models of the process and its actual physical state have to be defined depending on the considered level of abstraction (see in figure 2).

a) Case of measurable variables

Let us consider the case of a storage tank level evolution.

The events corresponding to "the storage tank is full", "the storage tank is empty" or "the storage tank level has reached the wished threshold" are significant for the upper levels but not necessarily for the lower one.

Reciprocally, events generated each variation of 20 dm3 of the storage tank level are insignificant for the upper levels but may be of importance for the lower one.

In order to respect the hierarchical consistency constraint, the process evolution statements for which the deduced pieces of information are significant for the considered level of the hierarchy have to be determined.

b) Case of unmeasured or non-measurable variables

Let us consider the processing time of the reaction1 (¤3). Its processing time may vary according to the current state of its components: amount and quality of raw materials, concentration of intermediate products, temperature and others factors of the environement [8]. It is often impossible to obtain relevant composition measurements, and even when it is not the case the delay is often too large for the data to be taken into account in real time [9]. So it is difficult to obtain information on the current physical state of the mixture, i.e. on the current state of the product, and in consequence to derive the activity duration.

The evaluation of the duration of this continuous activity may be done by simulation. This simulation utilizes the current state as initial condition and is based on a model of continuous nature as, for example, a set of algebraic equations or differential ones [10]. This type of continuous process simulator has been developed by ProSim with the simulation program ProSimBatch [11]. Including the simulator in the EG, the event generator allows to derive the date of the end-of-activity event (state event). In absence of end-of-activity sensor, this date is used to generate the corresponding event in the discrete event controller.

6. The modelling tool : Petri Nets (PN).

Petri Nets have been often used to model the discret aspect of dis-continuous processes [12], [13].

The classes of Petri Nets typically used depend on the hierarchical level, and on the point of view: performance evaluation (Timed PN), monitoring (Time PN, PN with fuzzy markings,...), or real-time local control (Grafcet).

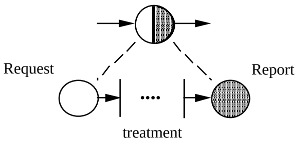
For example, models at upper levels of the hierarchy can be various kind of high-level PN (coloured PN, PN with objects,...).

To model dis-continuous processes, a few mixed approaches (continuous-discrete) have been developed as Hybrid Petri Nets [14] and Batches Petri Nets [15].

The possible use of various PN classes at each hierarchy level offers an important advantage for consistency (in both cases of modelling: real time control and simulation).

Let us show an example of discrete event modelling of the control policy for each level of the hierarchy, starting from the lowest level.

The command requests and command reports which are exchanged between a superior and an inferior level are sent by means of an object or a token depending on the class of PN which is used to model the considered level. A communication place is represented as shown in figure 4.

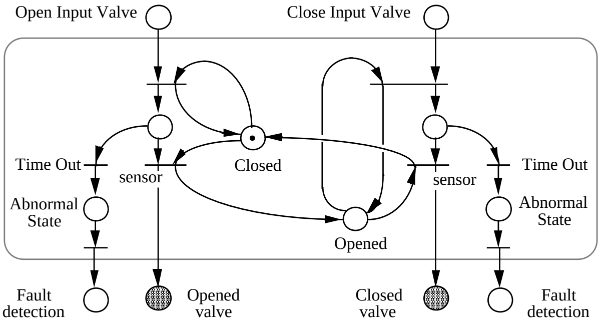


- Figure 4: Communication place -

6.1 *The local control modelling*

Control sequences are typically implemented by programmable logic controllers. They are often specified by means of Grafcet. This tool can be considered as an industrial standard based on safe interpreted PN [16].

The net in figure 5 represents the functioning of a charge valve to a vessel. It is considered as an elementary operation (level 1).



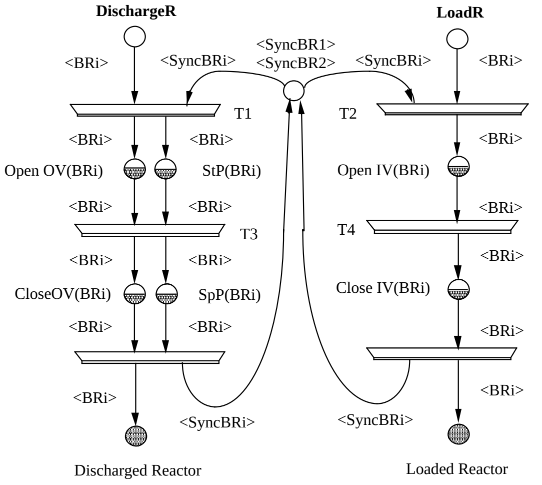
- Figure 5: control-monitoring net of a switch on/off valve -

6.2 *The service modelling*

The service modelling nets of a set of identical entities are the same (as for example the "load" function of different batch reactors (BR)). So, by introducing an object class (the object class <BRi> for the example given in figure 6), the set of these nets can be folded up on a unique Petri net with objects which controls the set of the identical entities [12].

The use of PN with objects (PNO) also allows to model different services on the same net as for example the heating and cooling functions of a reactor. Indeed, the heating function differs from the cooling one only by changing its parameters (assuming that the vessel has a three way valve to supply heating and cooling to the jacket).

Consider the example of the batch reactor services shown in figure 6: "LoadR" and "DischargeR". They respectively correspond to a BR filling and to a material transfer from a BR to another entity (emptying), without specifying the concerned BR entity. This specification is carried by the entering object.



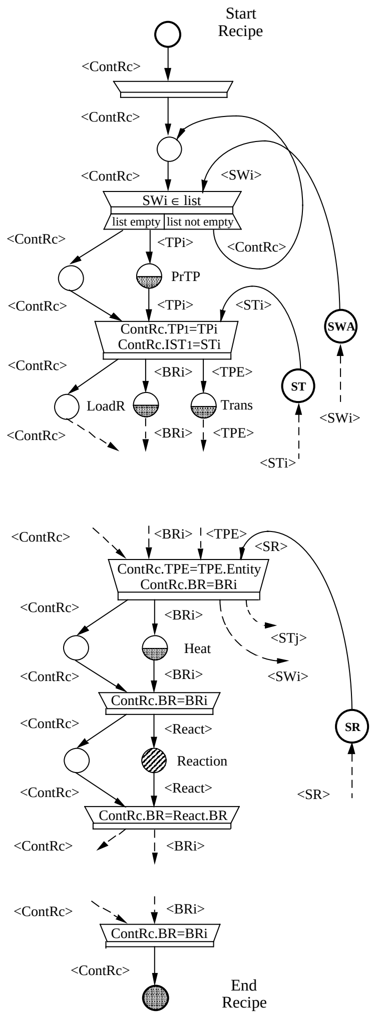
Object:
<SyncBRi>: synchronisation objects to ensure the exclusivity between the two services: <SyncBR1> or <SyncBR2>
Transitions:
T1, T2: SyncBRi=BRi
T3: BRi is empty
T4: Volume to be transfered in BRi is reached
Called operations:
CloseOV(BRi): Close Output Valve of the given reactor
Open IV(BRi): Open the Input Valve of the given reactor
StP(BRi): Start Pump of the given reactor
SpP(BRi): Stop Pump of the given reactor

- Figure 6: services modelling using PN -

6.3 *The recipe modelling*

As seen in ¤4, two types of recipes are used: the 'master' recipe and the 'control' recipe. The 'control' recipe is modelled by means of PNO. The 'master' recipe is represented by the structure of the net and by the predicates associated with its transitions. The 'control' recipe is an instanciation of the master recipe, it is denoted by the tokens location in the net. Thus for a 'master' recipe to be executed we just have to mark the input place of the net with the object corresponding to the given batch. It will be then updated in real-time.

The figure 7 shows a fragment of the model of the example control recipe (¤3), the net being too large to be entirely represented. The following points are shown: resources reservation, continuous treatment call (reaction) and some services call.



Control Recipe: **<ContRc>**

TPi: list of the required switching valves for the Transfer Path i
ISTi: Input Storage Tank i
Qi: Quantity of raw material i
BR: Bacth Reactor
OST: Output Storage Tank
React: formula, parameters (times, temperatures,...)

Resources

SWA: Switching Valves SW1 and SW2
ST : Storage Tanks ST1,...,ST7
SR : Limited Capacity Resources (pressure,cooling water, heating water)

Called Services

PrTP: Prepare Transfer Path
LoadR: Load reactor (BRi)
Empty: Empty reactor (BRi)
Heat: Heat, Cool (BRi)
Trans: Transfer (STi->BRi)
LoadST: Load Storage Tank (STi)

Objects:

<STi>: Storage Tank i
<BRi>: Batch Reactor i
<React>: Reaction Parameters and associated Reactor
<TPi>: Transfer Path i
<TPE>: Transfer Parameters and associated Entity

- Figure 7: recipe modelling using PNO -

6.3.1 *The resources modelling*

Let us consider limited capacity resources shared by the two reactors BR1 and BR2: pressure, heating and cooling utilities. Let us assume that these resources cannot be used simultaneously by the two reactors and that they are considered as an indissociable resource due to the fact that a reaction needs to have all of them simultaneously. This indissociable resource is then modelled as a unique one (i.e. as one token or one object) as shown in figure 7.

The switching valves are separated into two subsets: the first one containing the switching valves which are above the reactors (i.e. SW1, SW2) and the other subset composed of those which are below the reactors (i.e. SW3). The <TPi> object indicates the switching valves which are required to establish the considered transfer path. For example, to transfer raw material from ST1 to BR1 only the switching valve SW1 is required, and for a transfer from ST2 to BR1 both SW1 and SW2 are required.

6.3.2 *The resource allocation problem*

As it is shown in the modelling example in figure 7, each recipe has to reserve its needed resources ’just in time’ in order to optimise the resource mapping. This ’just in time’ reservation can lead the process to a critical situation if the required resource is not available at the right time.

Indeed let consider the case of an achieved reaction (reaction1: A+B->C) in the example given in ¤3. The mixture physical properties can undesirably change with excessive residence time in the reactor BR1. The mixture has to be transfered in an intermediate storage tank (ST5) within a short time delay. In order to prevent such a critical situation, before processing the batch, the upper level has to predict if the required resources will be available at the estimated date. The prediction can be done using a simulator of the entire process in function of its current state. The simulation might be based on the PN model, as for example Timed PN. Doing so, the plant resources state at the estimated date (near-future resources mapping) can be evaluated.

Thus the upper level is able to make the decision of processing or not the batch, preventing to lead the process to a critical situation and avoiding the propagation of its effect on the whole plant process. The process is then controlled according to its current and predicted states.

7. Conclusion

The paper has shown that techniques developed in the field of discrete manufacturing systems cannot be directly extended to batch systems. However it seems possible to model their discrete aspect using a hierarchical Petri Net based approach. The possible use of various PN classes at each hierarchy level offers an important advantage for consistency.

The consistency between the discrete model evolution (modelling the control policy) and the real process evolution can be ensured using a continuous model of the process at the different levels of the hierarchy. The consistency constraint has also to be ensured between the evolution of the continuous model of the process and the real process evolution. This can be done by means of the event generator.

Futher work will be done on modelling the process and the transient phase of continuous transformations (for example the start up and shut down of a separation), using continuous models.

References

[1] M.P. Rakotson, C. Vercauter, J.C. Gentina "Analysis of so-called 'continuous-discrets' production systems and criteria for defining a hierarchy of the coordination/super-

vision level", Automation of mixed continuous and sequential processes, ADPM'92, january 1992.

[2] G.S. Joglekar, G.V. Reklaitis
"A simulator for batch and semi-continuous processes", Computer and Chemical Engineering, vol 8 n¡6, 1984.

[3] P. Sawyer
"Batch standards on the move", The Chemical Engineer, 30 july 1992.

[4] B. Daubas, A. Pages and H. Pingaud
"Combined Simulation of Hybrid Processes", IEEE Intern. Conf. on SMC, San Antonio, USA, October 1994.

[5] Decotignie J.D., Pleinevaux P.
"A survey on industrial communication networks", Ann. Telecommun. 48, n¡9-10, 1993.

[6] Kopetz H.
"Event triggered versus Time triggered Real-Time systems", Operating Systems of the 90s and beyond, International workshop Dagstuhl Castle, Germany, july 1991, A. Karshmer and J. Nehmer (eds), pgs 87-101, Springer Verlag.

[7] Kopetz H.
"Six difficult problems in the design of responsive systems", Dependable Computing and Fault-Tolerant Systems, vol7: Responsive Computer Systems, H. Kopetz & Y. Kakuda (eds), pgs 3-26, Springer Verlag.

[8] K. Onogi, Y. Nishimura, Y. Nakata, T. Inomata
"An on line operating control system for a class of combined batch semi-continuous processes", Journal of Chemical Engineering of Japan, vol 19 n¡6, 1986.

[9] D.W.T. Rippin
"Control of batch processes", IFAC Dynamics and Control of Chemical Reactors, DYCORD+'89, Maastricht, The Netherlands, 1989.

[10] H. Pingaud, B. Daubas, B. Koehret
"Design of a simulator for continuous, semi-continuous and discontinuous processes, an evolution towards computer integrated processing", Automation of mixed continuous and sequential processes, ADPM'92, january 1992.

[11] ProSimBatch, user manual ProSim S.A., Toulouse, France, 1993.

[12] H.M. Hanisch
"Coordination control modelling in Batch Production Systems by means of Petri Nets", Computers Chem. Eng., vol 16 n¡1, pp 1-10, 1992.

[13] E.C. Yamalidou, J.C. Kantor
"Modeling and Optimal Control of Discrete-Event of Chemical Processes using Petri Nets", Computers Chem. Eng., vol 15 n¡7, pp 503-519, 1991.

[14] H. Alla, J.B. Cavaille, J. Le Bail, G. Bel
"Batch Production Systems : A discrete-continuous approach using Hybrid Petri Nets"
Automation of mixed continuous and sequential processes, ADPM'92, january 1992.

[15] I. Demongodin, N. Audry, F. Prunet
"Batches Petri Nets", IEEE International Conference on Systems, Man and Cybernetics,
Le Touquet, France, October 1993.

[16] R. David, H. Alla
"Petri Nets and Grafcet : tools for modelling discrete event systems", Prentice Hall International, UK, 1992.