15th Triennial World Congress, Barcelona, Spain

A WAY TO IMPLEMENT SUPERVISORS FOR HOLONIC PRODUCTION UNITS

Edgar Chacón [*,1]

[*] *LaSDAI – Departamento de Computación. Escuela de Ingeniería de Sistemas. Universidad de Los Andes. Mérida, Venezuela*

Abstract: Integration of the management of production systems has been considered one of the most relevant aspects in the automation process. In this work, authors propose to use abstraction mechanisms of the dynamics of continuous systems, to represent the behavior of processes (hybrid systems), and in this way, to supervise whole process. This work describes the implementation of mechanisms to supervise and control the behavior of Autonomous Production Units in integrated production systems, allowing the scheduling on-line of continuous production processes.

Keywords: Hierarchical Automation Systems, Holonic Systems, Process Automation, Object Oriented Systems.

1. INTRODUCTION

The need of information integration in an enterprise to support all the functions inside the enterprise is well known. The information integration follows an enterprise model that describes the information ¤ow through different components and levels in an enterprise. Several authors expose different approaches to model the enterprise and to establish the information ¤ow among the different systems that support the control of transformation / generation processes (mechanical, chemical, extraction, electrical energy generation, etc.). In manufacturing industries, the **Y-CIM** describes level, functions and information ¤ows clearly (Scheer, 1991). For batch production industries, the Computer Integrated Process model (CIP) describes the information ¤ow; this model follows the Y-CIM proposal (Scheer, 1994). A scheduling model in process industries that follows that proposal is given in (Loos and Allweyer, 1998).

ISA proposes a standard for integrate control activities (execution) with the management systems in the ISA standard SP 95 (ISA, 2000), which follows the Purdue Enterprise Reference Architecture (PERA model) of the Purdue University (PERA, n.d.).

A new approach for manufacturing systems is the *holonic manufacturing systems*. By de£nition, a *holon* is an element that is able to control itself, taking into account the knowledge that it has about itself and the environment. For these systems, the management of a production unit is performed taking into account three key aspects described in PROSA (Brusel *et al.*, 1998) for *holons*: *order, product* and *resources*.

On the other hand, the CIMOSA enterprise integration model de£nes several modeling constructs for systems for a production process (Konsake and Zelm, 1999; Salvato *et al.*, 1999). Those constructs are: *Behavior*, *Information*, *Resource* and *Organization*. These elements must be provided by an integrated system in order to allow their communication and information exchanges.

In (Chacón *et al.*, 2001), it is proposed integration architecture to allow the management of Continuous Production Systems (CPS). Those systems have a hierarchical structure made by the aggregation of single production components. Each component “transforms” some raw material in intermediate or £nal products, using internal resources (equipment), an ex-

[1] Financial support Venezuela organism: CONICIT and CDCHT-ULA. Corresponding author. Av. Tulio Febres, Fac. de Ingeniería, Mérida, Venezuela. e-mail echacon@ing.ula.ve

ternal resources (energy, HR) and services, in accordance with a production method. The product transformation is called *process*. The basic element in that model, it is the production unit (PU); de£ning PU as an autonomous object, which is able to take internal decisions in order to accomplish an assigned mission by a coordination of different PU involved in the production organization. It is expected that those systems make an optimal usage of production resources.

To perform its tasks, a PU must knows the state of the production infrastructure, optimizes its usage by selecting an optimal con£guration of the production resources. The PU must be able to communicate its state to an upper level to perform the coordination among the PU components of the production organization. This information exchange also considers resources usage, production levels and material consumption to ensure the necessary services of support units.

The information exchange among the different decision taking systems is obtained using a *three axis model* that represents a production system. This model is shown in £gure 1.

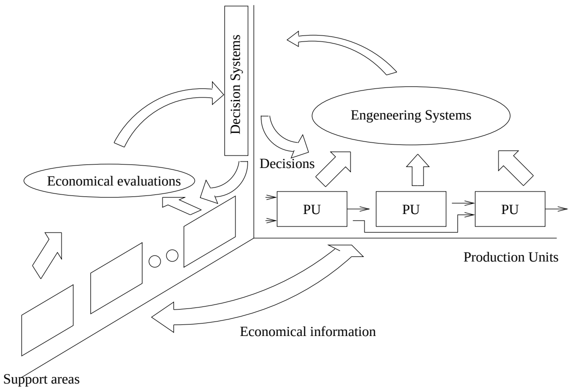

Fig. 1. Information ¤ow in Three-Axis Enterprise Model

The implementation of this model uses the holonic concept in order to have the whole information of each PU. To adapt this concept to the continuous product processes, we introduce a small variation of those three key components to the following ones:

- *Mission for a Production Unit*. A holon manages goals for a PU. Decisions are made by knowing the state of production resources and the best production method to obtain desired goal.
- *Production Resources*. A holonic system knows and manages the production resources.
- *Engineering aspects for production*. The optimal usage of resources to achieve a mission is obtained by the knowledge of production methods.

The information exchange among those concepts is given in £gure 2.

In practice, specialized software packages deal with the necessary applications to produce a production

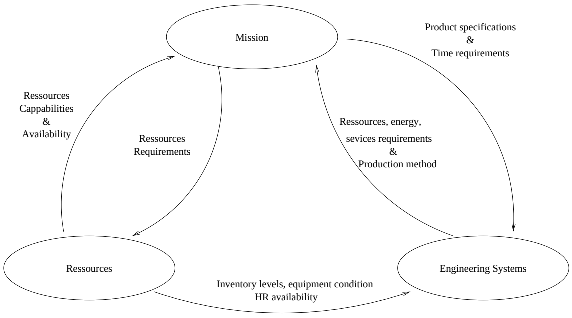

Fig. 2. Information exchange in a *holon*

plan, evaluate production methods and deal with the resources.

This work is devoted to achieve a generic description of a Production Unit: components and composite behavior of production units that are considered as a whole. An internal supervisor controls the behavior of the system by taking into account the behavior of the production process, the state of resources and the advance of its missions. The composed behavior can also been supervised by a hierarchical system where it belongs.

This work is organized in 6 sections including this one. In this section we give an introduction to the needs of have an integrated system able to take decisions based upon the knowledge of the supervised system. Section 2 describes brie¤y what are the principal logical components of a PU that make part of the PU and whose behavior determines the global behavior of the PU. In section 3 we introduce the way to build a global supervisor in order to manage the whole production process. Section 4, we describe the information and communication architecture that must support the integration system. The implementation approach to be used is described in section 5. Finally we give some conclusions in section 6

2. MODELING THE PRODUCTION UNIT

In order to obtain the model of a Production Unit, we describe the components of a PU, the internal behavior of each component, and the emergent behavior, which is obtained by a composition of the internal dynamics of each component. A schematic for the PU is shown in £gure 3.

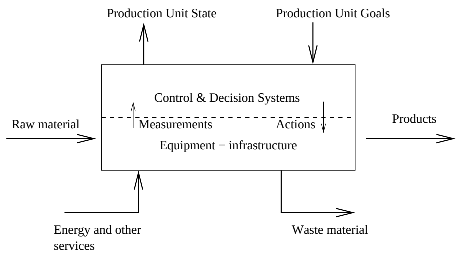

Fig. 3. Schematic of a Production Unit

2.1 *Logical Components of a Production Unit*

To understand the PU, as an autonomous system it is necessary to take into account the elements that perform or are related to the production process:

- **The Process**. The evolution of the process itself that indicates which is the advance in his mission. The transformation activities are mechanical or chemical.
- **The Resources**. Using a physical infrastructure, raw material and services that must be available in order to complete the production activities performs the processes.
- **Products**. Here, products represent raw material used by the process and £nal (intermediate) products obtained in the execution of the process.
- **The Production Method**. Each mission is performed using an organization (con£guration) of the PU infrastructure, a “recipe” that indicates quantities and qualities of the raw material, the con£guration and the set points for the process controllers.
- **The Supervisor**. The supervisor knows what is the mission for the PU and its advance, the state of resources, is able to select the optimal production method to achieve its mission or adapt a recommended “recipe” to its own capabilities.
- **The Integrator**. An agent is in charge to integrate the information exchange among the components of the PU. This exchange follows a set of business rules that describes how the elements interact.

The relations among the components is given in £gure 4 by using UML.

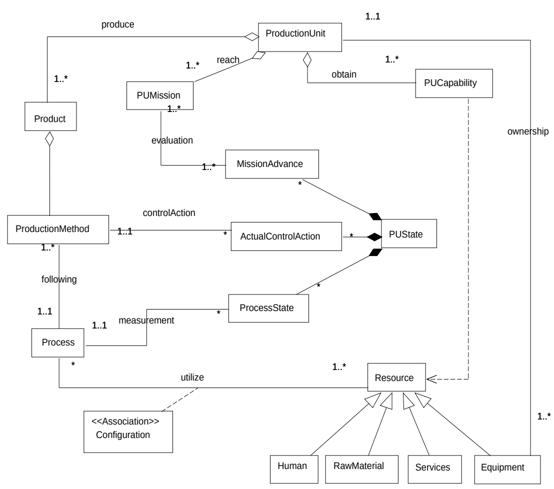

Fig. 4. UML model for a PU

2.2 *Behavioral Description of the Production Unit*

The behavior of a PU is a dynamic system that is described as a Discrete Event System. The whole behavior results form the composition of coupling the control system (supervisor) behavior and composite process behavior. Composite process behavior results from the states change of resources (process parts, raw material, energy, etc.) and the process itself.

Each component is modeled by a Petri net, which evolves according the presence of events. The evolution of a generic and simple PU, it is described by the Petri net in £gure 5. The transition nodes receive and send messages associated to events. Normal events are *Start a process* that start the production process considering that the process has all the necessary conditions to run it.

2.2.1. *Composing the process behavior* This Petri Net that model the behavior of the transformation process, it is not enough to represent the extended information in a PU, which allows taking decisions for the system that supervises the whole behavior of the PU. Other representative values associated to the whole behavior evolution are:

- *Capacity states*. Capacity is determined by the production method, capacity of the infrastructure (equipment), availability of resources. The evolution of the infrastructure varies according changes that can appears in infrastructure (Failures in components), changes in external resources (raw material) or changes in services (energy, waste treatment availability, etc.)
- *Available capacity*. It is the capacity non-committed in a production process. This allows increasing the production mission for the PU, or taking other missions.

The state of the PU results from the state of the different behaviors of the PU.

Behaviors are associated to the interfaces (communication channels). Principal elements are: mission, that is associated to the supervisor / coordinator of the PU; process state, which describes the evolution of the internal production process; resources state and PU con£guration. A set of mechanisms allows to external users to recover the state of a PU.

The following primitives establish the evolution of the PU.

- **Controllable events (commands)**
  - *Start*. Initiation of a production procedure to a PU.
  - *New Operation Mode*. Change the goal to a PU that has a task in execution.
  - *Shutdown*. Ending of a production task by a supervisor.
  - *Maintenance*. Execution of a maintenance activity for a PU. This maintenance activity can be due to a failure or by a maintenance plan.
- **no-controllable events**
  - *Task completed*

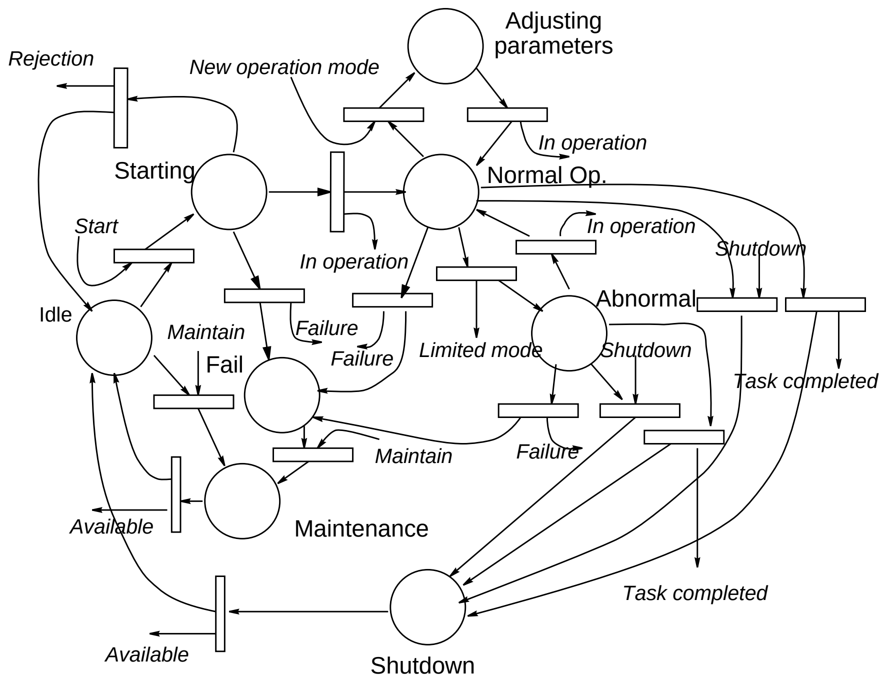

Fig. 5. Production Unit Behavior

- *Available*
- *Abnormal situation*
- *Failure*

2.2.2. *Discrete Event Dynamics of the Production Process* Continuous Production processes evolve according a set of physical - chemical laws that describes each kind of processes. This evolution can be described using hybrid systems (Lygeros, 1996), in such a way that will be described in this section. A local controller that ensures the right behavior of the system under speci£ed conditions controls the continuous dynamics.

A family of continuous controllers that has two functions controls the dynamic of the continuous system. The £rst one is to maintain the system under control in an operational region, the second one to move the system from an operational point to a new one.

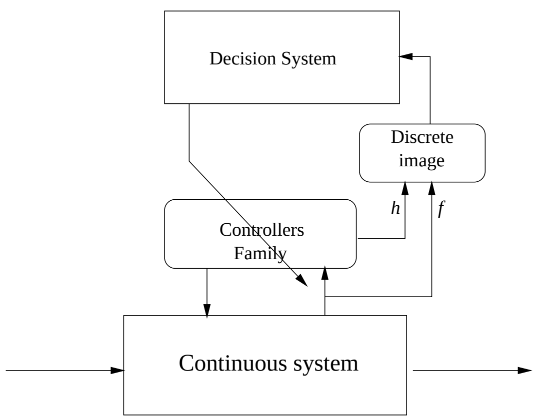

Fig. 6. Supervision Architecture

Several authors de£ne hybrid system as a system that mixes a continuous dynamic with discrete control, others uses as de£nition a continuous dynamic with jumps, which can be internal or imposed externally. We follow the de£nition given by Lygeros in (Lygeros, 1996), and we resume below. $HS = (X,U,Y,f,E,h,I)$

- $X = X_d \bigcup X_c$ states variables, where X_d is the subset of discrete variables and X_c is the set of continuous variables.
- $U = U_d \bigcup U_c$ set of inputs: discrete and continuous.
- $Y = Y_d \bigcup Y_c$ set of outputs: discrete and continuous.
- $f : X \times U \mapsto TX_c$ describes the continuous dynamics, where $q(t)$ is constant for the interval $(t^i_O, t^i_f]$ and x varies according:
$$\dot{x} = f(x(t), q(t), u(t))$$
- $E \subset X \times U \times X$ describes the discrete dynamics of the system by:
$$(q(t^i_f), x(t^i_f), u(t^i_f), q(t^{i+1}_0), x(t^{i+1}_0)) \in E$$
Changes in discrete variables happen at time t^i_f.
- $h : X \times U \mapsto Y$
- $I = (x(t_0), q(t_0))$ is the initial condition $x(t_0) \in X_d$ and $q(t_0) \in X_d$

In (Chacón *et al.*, n.d.) we can found a support to build a detector of the state of the continuous system under supervision. Two functions, f and h, must be considered in order to have the state of the continuous system. These two functions allow obtaining the discretized state of the process by make a projection of the continuous state over a discrete realization of the continuous process.

3. IMPLEMENTATION OF SUPERVISORS FOR A PRODUCTION UNIT

In (Lennartson *et al.*, 1996), the authors propose the following supervision structure that will be used by us to achieve the integration of the supervision to the global structure. See £gure 7

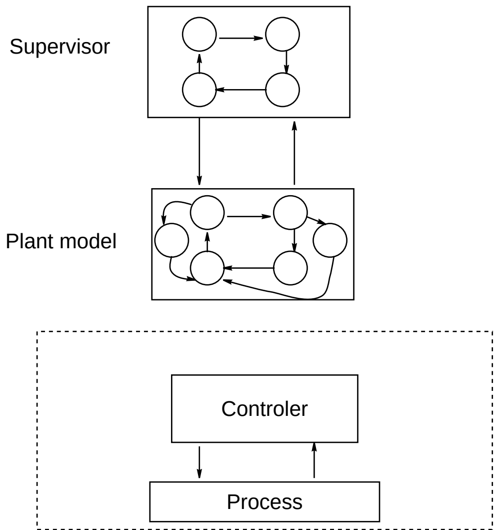

Fig. 7. Supervision of Hybrid Systems

Measuring the characteristics variables of the process and obtaining the events that change the state upon the discrete model of the process update the discrete state of the process. The model of the plant is built using the approach give in subsection 2.2.2

Sensors works periodically to obtain the events, when an event happens, the observer determines the new state.

In the same way that a simple PU evolves, a composite model represents the evolution of the production complex.

3.1 *Supervisors for integrated systems*

The whole process must be supervised in order to satisfy the system's mission. The supervision layer must drive the system to a state that satis£es the established mission, by selecting transitions that allows to the whole system evolves to the desired state. Figure 8 shows how the global behavior of the whole system is obtained by using supervisors that selects and enables the desired transitions.

Selection of transitions to be enabled are selected by the analysis of the resultant automata that describes the global behavior.

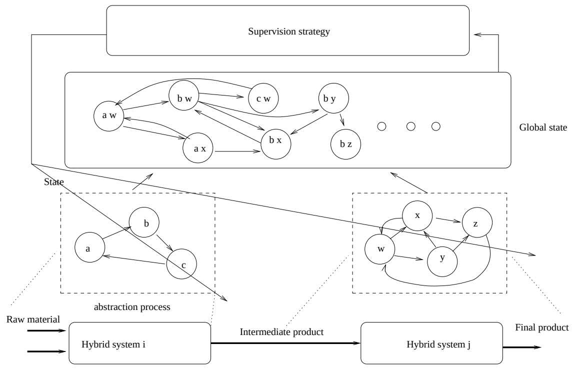

Fig. 8. Complex System's Evolution

4. INFORMATION AND COMMUNICATIONS TECHNOLOGIES

The control architecture that is proposed, in order to achieve the implementation of a supervision system is shown in £gure 9. The Information and Communication technology that supports the control architecture derives from SCADA's architectures already in place. RTU's must support classical control functions and incorporate mechanisms to detect events that are sent to the control centers. Events must be detected by applications on RTU and sent to control center, where the state of the production unit controlled by the RTU is updated. The information associated to each production unit is stored by using an object-oriented approach.

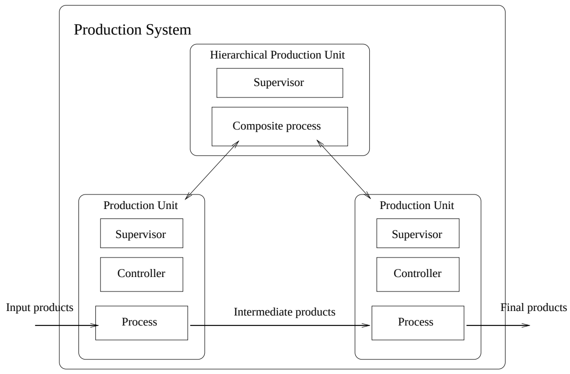

Fig. 9. Complex System's Supervision

4.1 *Production Unit Objects*

A Production Unit Class is an aggregation of other classes. A class describes behavior of PU, using the dynamical model described in subsection 2.2. Mechanisms for event detection allows maintaining updated the state of the Production Unit. Those mechanisms are distributed among the centralized system on a Control Center and the local control devices. The computer network used for the SCADA in place can be used without signi£cant changes.

The Production Units hierarchy is maintained at the control center, allowing coordination and scheduling activities. The behavior of the hierarchy is obtained by the composition of the elementary Production Units that belong to the composite Production Unit.

The implementation of supervisor allows giving recommendations to the operators, indicating the possible con£gurations of the system and the unacceptable ones.

5. ONGOING WORK

It is necessary to continue the development of this work, in order to build a system that helps to program people in fault detection and the evaluation of the reliability of composite systems. Also, it appears interesting to make more intelligent systems at the control level to detect the possibility of failures as an event, and in this way, to build a new program of activities.

The approach to be used in order to build the system is the construction of a set of applications that allows to collect the events' ocurrence in each one of Production Units. A SCADA systems, aready in place, collects all the data from the controllers. An interface allows to read the real time data base and detect the event ocurrence. A process' image follows the behavior of the PU's by using the behavioral model of each PU. The global behavior of the whole system will be obtained by using the exchange of events among the behavioral models of PU's and global system.

A kernel activates the eveolution of each model by accepting events. The programing language used for the prototype is C and uisng IPC system V. A graphical interface to show the behavior was built in java (Rivas, 2002). The new version wil be in java and JDBC to obtain the data of the SCADA system.

6. CONCLUSIONS

This works describes a method to build supervisors for distributed continuous production systems. Local supervisors, which is obtained, knows the performance of each production unit and informs about events to the control center which permit to follow the behavior of the local Production Units and, in this way, to obtain the global behavior of the composite system. The global supervisor evaluates the performance of the whole plant, detects incoherence on the system and gives recommendation to operators in order to change the optimal con£guration for a mission and Production Units states.

Acnowledgements. Author acknowledge the support of Venezuelan organism: CONICIT and the CDCHT of the Universidad de los Andes.

7. REFERENCES

Brusel, Hendrik Van, Jo Wyns, Paul Valckenaers, Luc Bongaerts and Patrik Peeters (1998). Reference architecture for holonic manufacturing systems: Prosa. *Computer In Industry* **37**, 255–274.

Chacón, E., I. Besembel, F. Narciso, J. Montilva and E. Colina (2001). An integration architecture for the automation of a continuous production complex. *ISA Transactions*.

Chacón, Edgar, Yanira Khodr and Gisela De Sarrazin (n.d.). Coupled dynamics for industrial systems. *Nonlinear Analysis* **47**(3), 1561 – 1570.

ISA (2000). *ANSI/ISA-95.00.01-2000 Enterprise - Control Systems Integration. Part 1: Models and Terminology*. ISA. Research Triangle Park, NC.

Konsake, Kurt and Martin Zelm (1999). Cimosa modelling processes. *Computers in Industry* pp. 141–153.

Lennartson, Bengy, Michael Tittus, Bo Egardt and Stefan Petterson (1996). Hybrid systems in process control. *IEEE Control Systems* pp. 45–56.

Loos, Peter and Thomas Allweyer (1998). Application of production planning and scheduling in the production industries. *Computer In Industry* **36**, 199–208.

Lygeros, John (1996). Hierarchical, Hybrid Control of Large Scale Systems. PhD thesis. University of California, Berkley. Electrical Engineering Department. University of California, Berkeley.

PERA (n.d.). Pera reference model for cim (isa publication). http://www.pera.net.

Rivas, Rafael (2002). Reducción de la complejidad en el control de sistemas a eventos discretos modelados mediante autómatas jerárquicos de estados £nitos. tesis de maestría. Technical report. Postgrado de Computación. Universidad de Los Andes. Facultad de Ingeniería. Mérida Venezuela.

Salvato, Guiseppe, Ion J. Leontaritis, Paul Wistone, Martin Zelm, Daniel Rivers-Moore and Daniele Salvato (1999). Presentation and exchange of business models with cimosa-xml. *Computers in Industry* pp. 125–139.

Scheer, August-Wilhelm (1991). *CIM: Towards the Factory of the Future*. Springer-Verlag. Berlin.

Scheer, August-Wilhelm (1994). Business process engineering: Reference models for industrial enterprises.