Coordination and Optimization in Oil & Gas Production Complexes

Edgar Chacón\textsuperscript{a,1}, Isabel Besembel\textsuperscript{b,2}, Jean Claude Hennet\textsuperscript{c,3}

\textsuperscript{a}LaSDAI – Departamento de Computación. Escuela de Ingeniería de Sistemas. Universidad de Los Andes. Mérida Venezuela

\textsuperscript{b}Gidyc – Departamento de Computación. Escuela de Ingeniería de Sistemas. Universidad de Los Andes. Mérida Venezuela

\textsuperscript{c}LAAS, Laboratoire d’Analyse et d’Architecture des Systèmes. 7 Av. du Colonel Roche. Toulouse - France

Abstract

In continuous production networks such as oil fields, global optimization requires to optimize not only production mechanisms, but also scheduling mechanisms for continuous flows of products through the different units that belong to the production complex. Two functions must be performed in order to efficiently operate a production network: coordination to ensure coherence and consistency of states, controls and decisions; and selection of the best configuration and set-points to run the process according to optimization criteria. The hybrid nature of this problem stems from the continuous processes involved and the use of discrete decision mechanisms to supervise the global behavior of the production process. To increase reactivity and reduce complexity in the global control scheme, each component of the system is described as a holon and the complex forms a holarchy. Each holon interacts with the other holons to contribute to the system’s optimization. Interaction is based on negotiation processes performed by agents that represent each holon.

Key words:

\textsuperscript{1} Financial support by Venezuelan organisms: FONACIT and CDCHT-ULA. Corresponding author. Av. Tulio Febres, Fac. de Ingeniería, Mérida, Venezuela. e-mail echacon@ing.ula.ve

\textsuperscript{2} e-mail ibc@ing.ula.ve

\textsuperscript{3} e-mail hennet@laas.fr
1 Introduction

Online scheduling in production networks have to take into account external requirements and internal conditions. Typical external requirements are: expected demand, market prices and rules, product specification in terms of quality and quantity. Typical internal conditions are: raw material availability, human resources, production costs, equipment availability and constraints, performance of production units, production methods, etc.

External requirements may vary as a result of changes occurring in the market. Internal conditions are subject to changes such as input variations (quality, volume), equipment failures, fluctuating supplies of raw material and energy. Coordination must ensure correct complying with production requirements while maintaining the process safety and optimization under the current conditions. Several methods have been proposed for automated scheduling in chemical processes. To deal with the complexity of such systems the proposed model range from purely mathematical models to mixed models [6]. In such complex systems each component can be considered as a subsystem responsible for its own processes and internal resources, but that has to meet external requirements. The requirements are defined in the global compromise that comes from the negotiation on capability and availability. This approach seems particularly appropriate for the management of oil production networks.

In oil production, several methods can be used to describe the characteristics of petroleum reservoirs and wells. As these characteristics vary on time, the models and parameters must be evaluated and estimated periodically in order to update production methods and control parameters. On the other hand, each well requires services and inputs in order to produce. After extraction, the petroleum flows through specialized plants that separate secondary products (gas, sand, water) and transport all them for final treatment, delivery or disposal. Figure 1 shows a general scheme for oil production. Generally, each one of the sub-processes is controlled and supervised by a specialized system, which exchanges information related to the state of sub-processes.

A classical approach to manage the whole production process consists in separating the functions into several levels that must work in an integrated way. In an upwards perspective these levels are: regulatory control, supervision and optimization, as shown in figure 2. In the context of a pyramidal architecture, it is generally assumed that the top level integrates functions such as: analysis, performance evaluation, optimization of the production process, scheduling of activities, and detection of abnormal situations (failures, inconsistencies between set points). In such an approach, lower levels do not process the information locally. They send all the information to a centralized system which determines the real state of the process and optimizes scheduling activities.
The global operational goal of the production system is to maximize the production benefit, provided that all the restrictions and safety conditions are maintained. Therefore, the production schedule is one of the most important results of this optimization. This schedule should be constructed using the key information regarding the current running conditions of the different units. This information should report in a more or less detailed manner the local parameters such as: state and capabilities of production units, financial conditions, inputs and product. The combination of this data with the information about the market should determine the ideal configuration for the system. Scheduling activities must be continuously made in accordance to the goals.
and the real state of the production units. Based on what has been said, these activities must be implemented locally in a reactive way, over the existent computer infrastructure. On the other hand, some centralization of information is required to run a set of applications, which are part of the whole process optimization (global optimization); as shown in figure 3. Those functions for oil production are:

- **Accounting functions:**
  - Oil production by wells, flow stations, regions, reservoirs, tanks.
  - Oil quality for each well and flow station.
  - Gas balance for each well, flow station, gas plant.
  - Services by well, flow station, gas plant.
  - Performance of each system.
- **Maintenance and inspection functions**
  - Equipment condition and performance.
  - Parts availability.
  - Operational conditions.
- **Process evaluation**
  - Process availability
  - Process criticality
- **Production planning and scheduling**
  - Resource allocation.
  - Expected production evaluation
  - Optimization

![Production process diagram](image)

**Fig. 3. Multiple views of a production process**

A totally centralized approach would require a large amount of information and communication, in order to collect all the data on the real condition of the process. Such an approach would be impractical and somewhat unrealistic. This global optimization and coordination determines the centralization of the decision making process.

The Information and Telecomunication architecture developed uses:

- Local controllers that are designed and implemented specifically for the characteristics of controlled processes. There are several common characteristics for control devices that allow to establish common protocols for
communication and their management.

- Supervision systems. Industrial protocols for SCADA allow the information exchange among remote control devices and control centers, and middleware systems communicate control centers to a production control/optimization level. Those communication services are mechanisms that allow the integration of information in control devices to a complete data integration scheme. The computer infrastructure is composed by the control computers, computers for supervision and general computers linked by computers networks with proper characteristics based on the communications capabilities of the computers and control devices and the requirements on communications (size, frequency and time restrictions of transmitted packets).

- A generic control equipment and computer architecture found for the automation of complex systems, which support the management of the complex.

In the field of automation, technical evolution leads to achieving control and safety decisions through local control devices. The local controllers are generally designed and implemented in specific agreement with the characteristics of controlled processes. There are several common characteristics for control devices that allow to establish common protocols for communication and their management. See figure 4 for a description of the IT architecture.

![Diagram of a control equipment architecture for complex production systems](image)

**Fig. 4. A control equipment architecture for complex production systems**

A possible way to solve the contradiction between distribution and integration is through aggregation/disaggregation mechanisms. A decomposition of the whole model as an aggregation of models that represent the behavior of processes and production units would greatly simplify the decision making process. However, an aggregation process always involves a loss of information.
Therefore, decision processes based on aggregated models carry on the risk of not leading to feasible actions. As in the manufacturing field [13], constructing robust plans and using local degrees of freedom can limit such drawbacks. The need for local flexibility has led to the introduction of the concept of Autonomous Production Unit [3], which is derived from the concept of holon [1,18,10] that is an element that is able of self-management and of informing the other units of its own state.

Production units are denoted PUs. They are considered autonomous because they are able to take their own decisions, by means of their local control devices, in order to keep the system safe and working according to specifications received from the supervisor. The specifications received from the supervisor result from the evaluation of the state of the subsystems, in terms of capability, performance, availability, relations among them, production strategies and external factors. Hence, several performance evaluation models are generally required. Their models and data should be coordinated.

For an oil production process, typical optimization variables are the oil volumes for each well. The main constraints express the expected production from each reservoir, the equipment topology, oil stocks, economical constraints and properties of facilities: flow stations availability and capabilities, pipelines, storage systems and distribution systems, taking into account capabilities and availability of each one of the components. Internal changes in one of these components may modify the production scheduling efficiency, and a reconfiguration of the currently applied production policy should be performed.

The distributed nature of the production system imposes an aggregation of components, which can be controlled independently, but must cooperate to achieve a global production goal. Several levels of hierarchical coordination are necessary, from the tactical upper level to the most technical lower levels. In particular, the network of pipelines that links the components should be organized and controlled. Each node receives and generates products as inputs and outputs in transformation, storage, generation, and delivering processes as is represented in figure 5.

![Fig. 5. A production Holarchy](image)

The aim of this work is to show how to define negotiation mechanisms that
allow for a negotiation among the PU’s, in order to establish common production goals, taking into account the fact that each PU has a certain degree of autonomy. The negotiation pattern should also be consistent with the current technological control architecture, in order to minimize costs and time for implementation.

This paper is organized into six sections, including the introduction and the conclusion. Section 2 shows the modeling process and its application to the case study. Section 3 is devoted to describing the dynamics of the process. The control and decision layer is presented in section 4, including negotiation agents. Section 5 is dedicated to implementation of supervision and negotiation functions within the industrial production complex.

2 Modeling the process

The system topology is represented as a hierarchical graph that describes the components of the system within its physical infrastructure. Each node represents a physical / chemical process and has a dynamic evolution (changes in values of internal attributes). The Discrete Dynamic System represents resource availability, production goals, production methods and simplified dynamics for continuous processes. These elements are analyzed as the main components of a holonic system [1]. A UML [14,15] description for a PU is given on figure 6, where object classes are represented by rectangles and the relationships among them by different types of lines, as is shown in figure’s legend.

![UML model for a Production Unit](image)

Fig. 6. UML model for a Production Unit

- The Production Methods class has a set of possible configurations for dif-
different goals according to resource conditions.

- The **Resource class** has the information on the state of the resource.
- The **Supervision and Control class** controls the evolution of the operation currently running on the resources, according to a production method, in order to satisfy a production goal.
- The organization of the PU is defined by a **configuration class**, which defines the physical configuration of the equipment, the objectives, the production method and, the control set points to be used.

For a given configuration, a PU has different capabilities and capacities.

### 2.1 Modelling the system static configuration

Each node of the system graph represents a set of physical components (production equipment) linked by pipelines according to some configuration. The **Configuration Class** to be used is selected by a local supervisor. This configuration includes: the organization of resources, the set of operations to be accomplished, and their operational mode (physical links of the infrastructure, resources, set points). Description of the configuration is obtained from the production methods package through a local decision system. Human operators, decision support system, and control systems compose the decision system. The control element associated to the **Supervision and Control class** in figure 6 observes the behavior and the state of the process, so as to perform its control, and to inform the operators about failures or errors in the process. The model for the supervision and control class is given on figure 7.
2.1.1 Resource class

The complete system can be viewed as an aggregation of different processes such as: transformation, mixing, transportation, and buffering processes. Specialized equipments perform those processes. They can be represented by mathematical models that describing their dynamics and performance. Other kinds of resources, such as energy and services, are necessary to perform the production activities. Some production costs are associated with energy consumption and energy usage, and these resources impose constraints onto the production process. Finally, some raw materials have to be transformed into intermediate or final products. Supply and use of raw materials also impose constraints to the process and have associated costs. The equipment condition varies according to the process mode. The resource description class receives information on the state of production resources and informs the supervision system about this state. Figure 8 shows a model for the resource class. Resources may be classified into the following types:

- Equipments to perform
  - Transformation operations
    - In transformation operations, several inputs are transformed into one or more outputs. Such operations are executed in transformation systems such as reactors, separators.
  - Buffering operations
    - The role of these operations is to store products. Buffering systems are vessels, tanks, etc.
  - Transport operations
    - Transportation systems are able to move the products along certain trajectories and within some limits.

- Raw materials

- Services
  - Energy.
  - Waste disposal.

2.1.2 Production methods and products

Several models for products and their evolution have been proposed in the field of manufacturing. In continuous production systems, products are obtained through processes that specify components and set points for different operational conditions. Changes in components impose changes in operational conditions and vice-versa. We propose an extension of the product model [7] to the continuous case, as shown on figure 9.
2.1.3 The configuration class

The *Configuration class* describes what resources must be used and how they should be organized in order to perform all the operations of a *production order*. Each operation requires some particular resources, and should produce intermediate products for other operations. Some operations can be executed in parallel, which means that resources must be organized physically (pipelines connections between production nodes and productions goals) for a time period. Each operational node has its own set-points that are coherent with the other operation nodes. See figure 10.

2.1.4 The complete model

The complete model of a PU is shown on figure 11 and a description of each element belonging to the PU follow:
Fig. 10. UML model for the Configuration class

Fig. 11. Static configuration of a holonic PU
2.2 The case study: an oil production system

The model of the oil production system is obtained by aggregation of elementary models of elements such as wells, production groups (wells, flow station), production regions. An object model of a well is shown on figure 12.

A production group consists of a flow station and wells. One or more production groups compose a production region. The static model for a production group is shown on figure 13.
3 The dynamics of the process

Each PU should operate in an autonomous way. However, the global system should also have a consistent behavior. In order to achieve these goals, each PU’s controller receives information about the process mission, objectives and goals, regulates the process, and sends information to the external system in order to combine and negotiate the goals. Figure 14 represents a PU as a node in the production network. This node contains a physical part, which performs the processes, and Decision and Information components.

![Diagram of Production Unit]

**Fig. 14. Production Unit description**

- **Physical system.** The physical system is composed of the equipment that performs all the physical production activities, using raw inputs, energy, etc. In a more general view, a physical process can be performed by a whole plant, which coordinates the activity of a set of physical subsystems.
- **Decision & Information system** The logical system which manages the plant uses information and knowledge on the physical process. It also has the capability to establish communication with others Decision & Information units belonging to others PUs. This logical system is composed of a set of Agents responsible for the implementation of the Decision & Information functionalities.

3.1 Behavioral Description of the Production Unit

The behavior of a PU can be described as a dynamic Discrete Event System. The whole behavior results form the composition and coupling of the discrete control system (supervisor) behavior and of the hybrid (discrete and continuous) process behavior. The hybrid (or composite) process behavior results from discrete state evolutions of resources (arrival and departure of process parts, raw materials, energy, etc.) and the evolution of the process itself.
The global discrete behavior of a PU can be modeled by a Petri net, whose tokens evolve as a result of the occurring events. The evolution of a generic and simple PU is described by the Petri net of figure 15. The transition nodes receive and send messages associated to events. Among the normal events, the process starting event is activated to start the production process if all the necessary conditions to run it are satisfied. Description of transition and place nodes are given in table 1.

3.1.1 Composing the process behavior

The Petri net of figure 15 models the behavior of the system for a typical transformation process. However, it does not carry all the information related to the PU. In particular, the whole behavior of the PU must be supervised and
Table 1
A Petri net description of the behavior of a production unit

decisions must be taken on the basis of the values of parameters or attributes such as:

- **Capacity levels.** A nominal capacity is fixed by the production method. However, capacity of the infrastructure (equipment) and availability of resources may change in time. Some changes may occur in the infrastructure (Failures in components), in external resources (supply in raw materials) or in services (energy supply, waste treatment availability, etc.)
- **Available capacity levels.** The capacity of a resource may be simultaneously assigned to several concurrent production processes. This may be the case if the PU has several missions.

A whole description of the PU state then results from its partial states related to its different behavioral levels. Current and projected behaviors are communicated through the interfaces (communication channels). Their main elements are:

- mission, that is determined by the supervisor / coordinator of the PU;
- process state, which describes the evolution of the internal production process;
- resources state
- PU configuration.

A set of mechanisms allows external users to capture the state of a PU. The following characteristics are used to describe the evolution of the PU.

- **Controllable events (commands)**
  - *Start.* Initiation of a production process within a PU.
  - *New Operation Mode.* Changing the goal of a PU currently executing task.
  - *Shutdown.* Ending 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 to a maintenance plan.

- **Uncontrollable events**
  - Task completed
  - Available
  - Abnormal situation
  - Failure

### 3.1.2 Discrete Event Dynamics of the Production Process

Continuous production processes evolve according to a set of physical-chemical laws that describe each type of process. The evolution of a PU can then be modelled as a hybrid system [12], in such a way will be described next. A local controller controls the continuous dynamics to ensure a correct behavior of the system under specified conditions.

There exists a family of continuous controllers to control the dynamics of the continuous system. They perform two functions. The first one is to maintain the system under control in an operational region, the second one to move the system from an operational region to a new one.

![Supervision Architecture](image)

Fig. 16. Supervision Architecture

Several authors define hybrid systems as systems that mix continuous dynamics with discrete controls, others define them as continuous dynamic systems with jumps, which can be internal or imposed externally. We follow the definition given by Lygeros in [12], which is summarized below.

\[
HS = (X, U, Y, f, E, h, I)
\]

- \( X = X_d \cup X_c \) states variables, where \( X_d \) is the subset of discrete variables and \( X_c \) is the set of continuous variables.
- \( U = U_d \cup U_c \) set of inputs: discrete and continuous.
• $Y = Y_d \cup Y_c$ 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 $(t_{i0}^i, t_{i1}^i)$ and $x$ varies according to:

\[
\dot{x} = f(x(t), q(t), u(t))
\]

• $E \subset X \times U \times X$ describes the discrete dynamics of the system by:

\[(q(t_i^j), x(t_i^j), u(t_i^j), q(t_{i+1}^j), x(t_{i+1}^j)) \in E\]

Changes in discrete variables occur at times $t_i^j$.

• $q(t)$ represents a set of discrete variables and continuous variables that remain constant for each discrete state $q$ of the system.

• $h : X \times U \mapsto Y$

• $I = (x(t_0), q(t_0))$ is the initial condition $x(t_0) \in X_d$ and $q(t_0) \in X_d$

The problem of detecting the discrete state of the continuous system in view of its supervision can be analyzed as in [4]. Two functions, $f$ and $h$, must be considered in order to construct that state for the continuous system. These two functions allow obtaining the discrete state of the process through a projection of the continuous state over a discrete realization of the continuous process.

### 4 The Control & Decision layer

According to the PU description of figure 14, production Units are split into their two components: Physical process and Decision & Information. The physical process implements the production process, which transforms its physical inputs into pre-specified, outputs. The Decision & Information component is the logical system in charge of driving the physical system towards the achievement of its goal. Such an action generally requires interactions with operators and with external systems.

#### 4.1 Intelligent Agents

The implementation of the Decision & Information layer can be achieved through the use of smart agents, which are able to control the internal system, make decision about the best way to ensure the production goals and exchange information with the environment where the PU is located. Some agents definitions follow.

“An agent is an encapsulated computer system that is situated in some
environment and that is capable of flexible and autonomous actions in that environment in order to meet its design objectives” [17].

“Agents are clearly identifiable problem solving entities with well-defined boundaries and interfaces, situated in a particular environment, designed to fulfill a specific purpose, autonomous, reactive, and proactive” [9].

Agents can be viewed as “an interacting decision maker that is able to proactively achieve its goals, while it is adapting to its dynamic environment” [2]. In order to specify the decision network where agents are involved. Each decision task should then be specified as follows:

(1) A set of requirements:
   (a) An input that contains the information to make decisions,
   (b) A decision space saying the set of possible choices,
   (c) A decision rule indicating how to make the decision, and
   (d) The activate condition to fire the decision (trigger).
(2) The decision network always meets the agent-oriented criteria,
(3) a possible reorganization of the decision network is done without changing the semantics of the decision process,
(4) there are multiple decision tasks in the decision network,
   the decision process is dynamic, and
(5) the involved decisions in the network are, at least, partly independent.

According to Zambonelli, Jennings, and Wooldridge in [19], there are two main types of multiple agent systems:

(1) Distributed problem solving systems, in which agents are explicitly designed to cooperatively achieve a given goal. All agents are known a priori, and all agents are supposed to be benevolent to each other and they can trust one another during interactions.
(2) Open systems in which agents, not necessarily co-designed to solve a common goal, can dynamically leave and enter the system. At the dynamic arrival of unknown agents, their needs must be taken into account, as well as the possibility of self-interested behavior in the course of interactions.

This study considers the first type of multiple agent systems, where each agent is assigned a specific role in the system, that is, a well defined task or responsibility in the context of the overall system, that the agent has to accomplish in an autonomous fashion, without any centralized control.

Interactions are identified and localized in the definition of the role itself, and they help characterizing the position of the agent in the organization.

(1) Each agent becomes a separate source of action, striving to accomplish its role and being fully responsible for it.
(2) Since agents typically embed most of the functionality they need to ac-
complish their role, inter-dependencies among systems components are reduced.

An agent can play one or more roles. Agents typically need to interact with each other in order to exchange knowledge and coordinate their activities. This interaction occurs via sensors and actuators, which are mechanisms that enable agents to sense and act on a selected portion of the environment.

A global coordinator agent can then be introduced. He is in charge of supervising and mediating the interactions for all the other agents, leading to a hierarchical organization.

Organizational rules (OR) express relationships and constraints among roles, among protocols, and between roles and protocols, that can drive the identification of the organizational structure. As pointed out by Zambonelli et al. in [19], “OR can be considered as liveness and the safety properties of the organization”.

4.1.1 Temporal intelligent agents

As pointed out by Dix, Kraus and Subrahmanian in [5], an agent program or agent, in short, is continuously engaged in an “event occurs-think-act-event occurs” cycle, that suggest the use of some kind of support for temporality. A temporal agent program (TAP) is an agent that executes actions that have duration, recording some of its state changes during action execution in a temporal status set defined by a set of checkpoints. “When an agents state changes, the agent must find a new temporal status set that satisfies various semantical requirements. This tells the agent what it is obliged to do now and in the future, what it is permitted to do now and in the future, what it is forbidden from doing now and in the future, and also allows it to determine what in fact it will do”

TAP permits to have rules of the type of Event-Condition-Action (ECA) rules, which allow temporal indeterminacy, and also permits rules of the type utilized in logic programs and deductive databases. In our case, we are interested in supporting ECA rules in the context of active databases, which are more suitable for treating business rules (BR) covering the aforementioned OR, in continuous production complexes (CPC). This type of rules has the form: “if condition C is true and some event happens then do an specific action”. At this time, we do not consider temporal indeterminacy in rules. For the example of an oil production well, a rule is like:

if (process in pp3) and (event in resource) then
    evaluate ((t4), (t5) and (t6))
4.1.2 Agent negotiation

Negotiation is considered the most fundamental and powerful mechanism for managing inter-agent dependencies at run-time [8]. Jenning defines negotiation as “the process by which a group of agents comes to mutually acceptable agreement on some matter”. This concept is so central precisely because agents are autonomous.

4.2 Negotiation

In our approach, each PU has an associated agent that is in charge of negotiating with other agents of the same level, and with other agents in the next below and/or upper level in the hierarchy of PUs composing the CPC. The PU agent can negotiate about its resources and products, by following a specific goal of the PU, that is involved in a specific objective of the holarchy [16], and taking into account the PU global and current state of resources and its production capabilities. In this vision of the negotiation process, a PUs agent (PUA, in short) is a TAP able to make honest proposals, trading options, offering concessions, and either builds a mutually acceptable agreement or aborts the negotiation process.

4.2.1 A negotiation Process

Following Jennings in [8], the negotiation process deals with three main topics:

(1) Negotiation Protocols (NP) considered as a set of rules governing agents interaction, which cover the types and number of participants, the negotiation states, the events causing negotiation state changes, and the valid actions of each participants in each particular state.

(2) Negotiation Objects (NO) that are defined to either reach an agreement or abort the negotiation process. NOs have properties specifying negotiation issues, and also have operations that can be performed depending on the NP.

(3) A decision making model which is utilized to act in line with the NP in order to achieve PUs goals. This model is also expressed by using a set of rules. We consider that NOs are included in the Business Object class, and both NPs and the decision making model are included in the Business Rule class.

Our proposal is to consider the negotiation process as a semi-automated process at the higher levels of the holarchy. At the lower level, an automated negotiation is considered. This last type of negotiation is viewed in [8] “as a distributed search through a space of potential agreements” where NOs de-
fine the dimensionality and topology of the space, and both NPs and decision making models determine the direction of the search.

A negotiation process begins when an agent makes a proposal defining a portion of the space in which it is willing to make agreements. The process continues by other participant agents suggest changes on the proposal that modify search space making it more acceptable, and process terminates when it is reached an agreement or when the protocol dictates the impossibility of making it.

A first kind of negotiation is defined by the way an agent responds to a proposal. It can be a critique or a counterproposal. A critique is a response saying which parts of the proposal the agent likes or dislikes. It suggests some constraints on NOs, and also indicates acceptance/rejection of the proposal. A counterproposal is a proposal made in response to an original proposal normally containing changes on the agreement space. This kind of negotiation avoid the justification of a particular response of the agent, and the persuasion of other agents to perhaps completely change the agreement space. This is because responses are confined to NOs, and both justification and persuasion are involved by negotiation rules expressed by NP and the decision making model.

The second kind of negotiation includes the first one by permitting justification and persuasion. This is called argumentation based negotiation where it is included threats, rewards, and appeals. All of them modifies the agreement space with new conditions. Threats define that a failure in accepting this proposal means something negative will happen to the agent. On the contrary, rewards means something positive happen to the agent if it accept the proposal. Appeals mean the agent should prefer this alternative over that option for some reason.

The main problem with this last type of negotiation is the implementation complexity added to the agent. For this reason, we only consider here the first kind as an initial point to our TAP.

4.2.2 An illustrative Example

As an illustrative example, we select a production region, which has a subsoil electrical pumping production method. The production region is composed of a group of 10 wells, and one flow station. The capacity of the flow station and the properties of the reservoir determine the production amount for the whole system. Each well must produce at its maximal flow rate, which is limited by the pump and pipe conditions. An initial evaluation is computed by a mediator, assuming that the last condition is satisfied for each well and for the flow station.
The behavior of the well components is modelled by the Petri Net given in figure 17, and a description of the nodes is given in table 2.

### Table 2
A description of the Petri net for the behavior of the well

The global evolution of the well is described by a model similar to the model of a PU behavior given in figure 15. A mediator is located at the control center and has the most recent information about the capabilities of each well. It elaborates a proposal through multi-criteria optimization and sends it to each
one of the wells and to the flow station. The procedure for starting the well production requires acceptance of the proposed production level.

Each well receives a production level, which is evaluated internally, using the scheme given in subsection 5.3. According to the results of the evaluation, it sends a counter proposal or an agreement. Each well has an simple model of the reservoir that allows it to estimate its production capacity, taking into account the conditions on the pump and pipes. Models and conditions must be frequently updated at the control center.

For the rule mentioned in 4.1.1, evaluate \((t_4), (t_5) \text{ and } (t_6)\) is the result of evaluation for possible states for pipe, pump and reservoir than have a valid production method.

A rule showing a counterproposal is:

\[
\text{if (proposal can not be achieved) then} \\
\text{send counterproposal}
\]

5 Implementation of Supervisors for a Production Unit

5.1 Models for supervision

In [11], the authors have proposed an architecture of models for supervision sketched on figure 18. A similar modelling structure will be used here to achieve the integration of the supervision tasks into the global decision process.

![Supervision Model](image)

Fig. 18. The compound supervision model

On the basis of a model of the plant, built using the approach given in subsection 3.1.2, the supervision process is organized as follows:
• measure the characteristic variables of the process (sensors works periodically to observe the system),
• identify the observed events that induce changes of states in the discrete model,
• update the discrete state of the process.

Control and supervision tasks are executed in a cyclic way, as shown in fig. 19.

![Diagram of the control cycle](image)

**Fig. 19. The control cycle**

### 5.2 Supervision of the integrated system

In the same way that for each PU, a model of the production complex can be constructed to represent and supervise the evolution of the composite system. This model generates the first attempt to achieve the negotiation.

The whole process must be “supervised” in order to satisfy the global missions of the system. Using the compound discrete event model (an automaton or a Petri net), the supervision layer must drive the system to a state that corresponds to the completion of the mission, by selecting the transitions that allow the whole system to evolve to the desired state while optimizing one or several objective functions. Figure 20 shows how the global behavior of the whole system, obtained by using supervisors that select and enable the desired transitions.

### 5.3 A multi-attribute view of supervision

By essence, supervision is multi-attribute, since it corresponds to different views of production units and production processes such as is described in Introduction in figure 3. Normally, the enterprise is organized in services that cover all the relevant attributes for each process. Several sets of data can be constructed from the same source, the process, and several missions can be assigned to its achievement. Possible conflicts may occur. They can be solved
automatically through multi-objective optimization or negotiation between smart agents, or through human collaboration.

5.4 Negotiation: PUA architecture

The negotiation among PU’s, it is oriented to ensure the accomplishment of a global production goal, where several PU’s develop activities that support the global goal. Some of them are absolutely necessaries, and other one can give a contribution to obtain the goal. The global goal is accepted for the entire holon after an agreement among the different components of the PU.

The arrival of a goal change (or new goal) from the group of PU belonging to a upper holon, implies an evaluation of the feasibility of the task using internal resources and the promise of external resources and the existence of a method that can perform a set of tasks to achieve the goal. Once the evaluation is obtained, the PU sends an agreement or a counter proposal until obtain the agreement. This agreement is a goal for the PU, whose execution will be supervised.

Supervision cycles in this approach imply a continual evaluation of the accomplishment of an accepted goal. How in the above-mentioned approach, it implies resources evaluation and the control of the process, in order to establishes: finalization of the committed goal and the feasibility of that goal taking into account the resources condition. The state of resources is evaluated using different approaches according to each kind of resource. Supervisor detects incoherencies between resource’s state and current control law. Detection of incoherence implies the evaluation of the feasibility of the goal by using another control law, which will be selected from the possible methods. If it is not possible to have a new control law, the goal will be rejected, and the system
goes to a degraded operation mode, and wait for a new goal.

The inner PUA is composed of four processes. The first one is devoted to controlling the physical process, where Process Supervisor & Control and Configuration classes are contained. The second one is used to maintain resources updated, by using Resource and Configuration classes. These two processes transmit the current state of the physical process and the actual state of resources utilized in the physical process, to a third process, called the supervisor agent. This agent is a PUA mainly charged of the evaluation of the PU’s current state through Production Unit and Production Method classes. The supervisor PUA knows the PU’s mission and decide which production method to apply for reaching that mission. In order to do that, the supervisor PUA may be emit an order of negotiation to the fourth process, which is the negotiator PUA. This last process negotiates with others PU’s of the enterprise, in case of external resources of this PU are needed to accomplish its mission or PU mission is part of the general mission of the enterprise. Figure 21 shows the PUA architecture.

The negotiation process is presented in figure 22 by using a sequence diagram that explain a first approximation of messages interchange among business object.

5.5 Technologies for implementation

The Real Time Units (RTU) must support classical control functions and incorporate exchange mechanisms. In particular, events are detected and the information is sent to the control center where the state of the production unit controlled by the RTU is updated. The information associated to each production unit is collected and processed by the smart agents in charge of the PU. In the case of the oil production network, the information and communication technology that supports the control architecture derives from SCADA architectures already in place.
To achieve the implementation, we must make a map of the agents for each basic holon (wells, flows station), and place each one at the RTU or at the control center. Supervision agents, control agents and resources agents are placed at the RTU. Some agents can be using specialized software systems also located at the control center or at specialized equipment that help to evaluate pipe, pump and reservoir conditions. Negotiation agents, which exchange a lot of information to achieve an agreement with others negotiators, are placed at the control center. The construction of production group holons, and production region holons is achieved at the control center, and are derived from the basic holons, and they derive their state form the state of the basic holons. Figure 23 sketches the implementation to be installed at the control center. The negotiation schemas are the same for each level.

6 Conclusions

The generic approach which has been proposed applies to the integrated control and optimization of distributed production complexes. The necessary coordination between production units is achieved through a hierarchical organization of functions and through communication protocols to implement
optimized decisions, to insure safety and satisfaction of technical constraints. On the other hand, the global efficiency of such a distributed system requires autonomy and reactivity at the local level of production units. The solution proposed to combine autonomy with integration is through automated hierarchical negotiation. Production units and central services are represented by software agents that negotiate on task assignment, constraints and set-points. A key advantage of such a coordination mechanism is a drastic reduction in complexity of the models used at the central level and of the data flows between production units and the global coordination unit. The typical continuous production system considered in this paper is a distributed oil production complex. Implementation of coordination and optimization schemes is achieved through an architecture that uses: DES models to describe the whole internal dynamic behavior of the process; hybrid systems to control it; cooperative agents that evaluate the internal behavior of production units, their capabilities and capacities, and that are able to interact with each other to accomplish the objectives and goals of the system.

Fig. 23. IT architecture proposed to implement the system
References


**Edgar Chacón**  
Professor at the University Los Andes, Mérida Venezuela, he belongs to the research group on Discrete Systems and Industrial Automation. His main areas of interest are discrete event systems, hybrid hierarchical systems and integrated automation.

**Isabel Besembel**  
Professor at the University Los Andes, Mérida Venezuela, she belongs to the research group on Data and Knowledge Engineering. Her research interests are mainly related to data structures, data bases, object-oriented models, business rules and software agents.

**Jean Claude Hennet**  
Directeur de Recherche CNRS in the group on Organization and Control of Discrete Systems of LAAS-CNRS, Toulouse, France, his research interests are in the fields of system theory, optimization, control under constraints, discrete event decision systems and manufacturing systems.