



HAL
open science

H2CM: A holonic architecture for flexible hybrid control systems

Carlos Indriago, Olivier Cardin, Naly Rakoto-Ravalontsalama, Pierre Castagna, Edgar Chacòn

► **To cite this version:**

Carlos Indriago, Olivier Cardin, Naly Rakoto-Ravalontsalama, Pierre Castagna, Edgar Chacòn. H2CM: A holonic architecture for flexible hybrid control systems. *Computers in Industry*, 2016, 77 (Avril 2016), pp.15-28. 10.1016/j.compind.2015.12.005 . hal-01277310

HAL Id: hal-01277310

<https://imt-atlantique.hal.science/hal-01277310>

Submitted on 1 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

H²CM: A holonic architecture for flexible hybrid control systems

Carlos Indriago^a, Olivier Cardin^{a,*}, Naly Rakoto^b, Pierre Castagna^a, Edgar Chacòn^c

^aLUNAM Université, IUT de Nantes – Université de Nantes, IRCCyN UMR CNRS 6597 (Institut de Recherche en Communications et Cybernétique de Nantes), 2 avenue du Prof. Jean Rouxel, 44475 Carquefou, France

^bLUNAM Université, Ecole des Mines de Nantes, IRCCyN UMR CNRS 6597 (Institut de Recherche en Communications et Cybernétique de Nantes), 4 Rue Alfred Kastler, 44300 Nantes, France

^cDepartamento de Computacion, Escuela de Sistemas, Universidad de Los Andes, Mérida Venezuela

Several studies typically arise from the interaction of discrete planning algorithms or control and continuous processes, normally called hybrid control systems. It consists in three distinct levels, the controller, the plant and the interface. Hybrid control systems are conventionally modeled by switching patterns using the whole system instead of atomic resource. Therefore, the reconfiguration process is complex because it must take into account the system as a whole, making the hybrid control systems inflexible and more susceptible to uncertainties. The need for flexibility thus leads several teams to investigate the application of holonic paradigm to hybrid control systems. The objective of this paper is to demonstrate the possibility to apply almost directly a holonic discrete-event based reference architecture to hybrid control systems. A case study of industrial electricity generation process was taken, specifically a combined cycle plant (CCP) for verifying the proper operation of the proposed architecture.

1. Introduction

Manufacturing systems are traditionally divided into three main classes, namely discrete-event, continuous and semi-continuous processes [1]. Discrete processes class deals with systems where the individual parts are produced using various discrete loosely coupled operations (such as machining, drilling or grinding), then pieced together in an assembly line to create the main end-product. The control of these kinds of systems is classically optimized via scheduling techniques, able to determine the best order of parts to manufacture in order to maximize a given criteria, such as makespan or mean tardiness. However, this fully predictive approach is progressively replaced by predictive-reactive approaches, able to react to disturbances appearing when applying the schedule on the shop floor [2]. Mehta and Uzsoy [3] classified these approaches into four main categories: completely reactive approaches, predictive reactive approaches, robust scheduling and knowledge-based scheduling. Another explored way is to redefine the control architecture in order to give autonomy of action and decision to

specific entities. This concept is usually denoted as holonic paradigm, creating holonic manufacturing systems (HMS) [4]. Holonic control architectures are flexible architectures which allow online reconfiguration of processes. A holonic manufacturing architecture shall enable easy (self-)configuration, easy extension and modification of the system, and allow more flexibility and a larger decision space for higher control levels [5].

The second class deals with continuous processes. It involves continuous flow of materials such as water or bulk chemical for example and utilities through process units interconnected via piping streams. Their behavior is usually described as a system of differential equations, describing the outputs of the system considering the inputs and the values of the parameters that can be addressed. The optimal control of the system generally consists in inverting the system, i.e. determining the best set of parameters in order to obtain the desired outputs considering the inputs.

Semi-continuous processes, belonging to third class, also involve continuous flow of materials and utilities, similarly to continuous processes, but are not operated with a purely steady-state mode. Several studies typically arise from the interaction of discrete planning algorithms and continuous processes, called hybrid control systems (HCS) [6].

* Corresponding author.

E-mail address: olivier.cardin@irccyn.ec-nantes.fr (O. Cardin).

The hybrid approach used in this paper is a HCS, which has a continuous-time linear time-invariant plant described by linear differential equations, which involve continuous valued variables that depend on continuous time. It is controlled by a discrete-time linear time-invariant plant described by linear differential equations, which involve continuous-valued variables that depend on discrete time [6]. According to [7], HCS consists in three distinct levels (Fig. 1). The controller is a discrete-state system, a sequential machine, seen as a discrete event system (DES). The controller receives, manipulates and outputs events represented by symbols. The plant is a continuous-state system typically modeled by differential equations and is the system to be controlled by the discrete-state controller. The plant receives, manipulates and outputs signals represented by real variables that are typically (piecewise) continuous. The controller and the plant communicate via the interface that translates plant outputs into symbols for the controller to use, and controller output symbols into command signals for the plant input. The interface can be seen as consisting of two subsystems: the generator that senses the plant outputs and generates symbols representing plant events, and the actuator that translates the controller symbolic commands into piecewise constant plant input signals.

The need for flexibility thus lead several teams to investigate the application of holonic paradigm to HCS [8–10]. The objective of these works is to develop a control architecture that would fit the specificity of hybrid systems. Therefore, they all focus on creating holons based on the classical functions to be executed on this kind of systems. Although efficient, these approaches lack of generality and disconnect the results from the set of results already available from the numerous studies in the field of discrete-event systems. The objective of this paper is to demonstrate the possibility to apply almost directly a holonic discrete-event based reference architecture to HCS.

In Section 2, a state of the art of holonic modeling in hybrid systems is provided. Then, a proposal of holonic modeling in HCS is detailed in Section 3. A study case is presented in Section 4, followed by the implementation of holonic paradigm before the conclusion and perspectives.

2. State of the art

2.1. Product resource order staff approach (PROSA)

PROSA is one of the most referenced holonic architectures. Unlike other reference architectures such as ADACOR [11], HCBA [12], ADACOR² [13] or ORCA-FMS [14], the holons defined in the

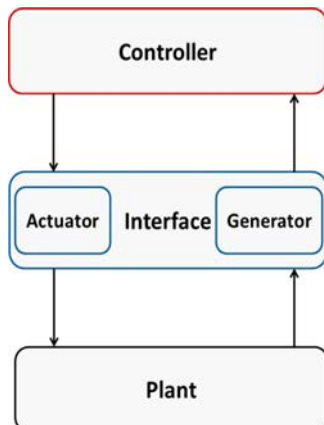


Fig. 1. Hybrid control system.

architecture are relatively generic and disconnected from the sole application to discrete manufacturing environments. This feature makes it a very good candidate for being the basis of an extension of the paradigm to a different class of systems. This is the main reason why this paper relies on PROSA only to illustrate the proposed concepts.

PROSA name derives from an acronym Product Resource Staff Order Approach [15]. The architecture consists of 4 basic holons that are the product holon, the order holon, resource holon and staff holon, see Fig. 2. The product holons (PH) deal with all the data related to the references produced in the HMS. They roughly act like a database for all the other holons of the system to enable them to access all the data (routings, bills of materials, etc.) needed to be able to produce the right products with a sufficient quality. The resource holons (RH) corresponds to physical entities able to act on products (such as factories, machines or tools) and contain all the data and knowledge to be able to control and organize these entities. The order holons (OH) handle the product during its manufacturing, deal with the logistics needed in the routing of the product. These holons interact with each other to achieve better production, thus a system of holons that can co-operate to achieve a goal or objective is calling holarchy. Finally is the staff holon, its function is to help basic holons to find a solution to difficult decisions, it plays the role of an external expert to basic holons. Therefore, the staff holon is meant to be part of complex decision making mechanisms. The aim of this article is to define the general framework in which those mechanisms would be implemented, but not to define the best ones in a specific case. Thus, the staff holon is not used in this paper.

2.2. Holonic approaches of hybrid systems control

The application of holonic systems on continuous systems has been studied by several authors. An implementation of holonic production system on a continuous system is for example proposed by [9]. In this work is showed from Table 1 that control problems of both a static and dynamic nature occur at every level of the manufacturing problem.

Although the algorithms and problem domain are obviously different, in each case there is a clear problem of regulating a control variable via feedback in order to align an output variable with a desired setting. Within a holonic context, the key unifying requirement in all of these different static and dynamic control algorithms is that they be goal seeking. That is, the control algorithm should be provided with a set of requirements and must first convert these requirements into a solvable problem before solving the problem. The goal seeking system depicted as in Fig. 3

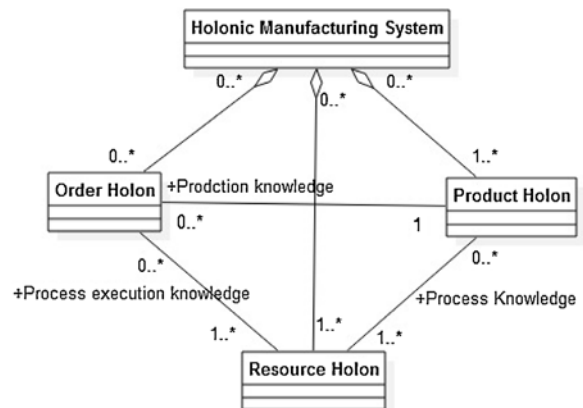


Fig. 2. Basic building blocks of a HMS and their relations, based on [16].

Table 1
Static and dynamic control problems on different manufacturing levels.

Type	Level	Static	Dynamic
Continuous	Local control	Model-based set up	Model-based control
Discrete/ continuous	Process control	Schedule	Dynamic rescheduling
Discrete	Process management	Production planning	Dynamic planning

has the feature that it internally converts goals or requirements into an allowable set of parameters or trajectories of behavior via some form of decision function, D . The actual control action is then carried out in process P . The terminology in Fig. 3 is as follows:

- P : process (model + controller);
- D : decision problem;
- Φ : decision principle;
- m : set of alternate decisions;
- u : set of alternate events;
- y : set of possible outcomes.

The development of appropriate (and flexible) decision problems is the key to embedding control activity within holonic systems whose goals may differ depending on the environment that they operate in. A distributed optimization based decision principle is well suited to holonic systems, and the goal-seeking methods could help with the integration of complex optimization based control algorithms into manufacturing operations – a relatively uncommon event at present.

Another holonic architecture developed for continuous systems is that proposed by [10]. It proposes a holonic structure based on a making decision system called production unit, see Fig. 4. This unit is the foundation for exploiting the fractal structure characteristic of holonic systems. The aggregation of production units is then called complex production system.

The production unit has physical (raw materials, finished goods, services) and logical (goals, state information) inputs and outputs. In addition, this production unit follows a holonic structure composed of three basic holons called resource holon, mission-product holon and engineering holon.

Finally, the work developed by [8] shows an approach to distributed coordination process for reconfigurable control. In this work, control architecture is developed using four basic elements such as the product element, the unit element, the header and the service element. Although it does not explicitly address to holonic paradigm, the goal of this architecture is to have a control system based on reconfigurable and flexible interaction models, similar to holonic systems and it can be implemented in continuous and semi-continuous process. The functionality of these individual types of process elements can be defined as follows:

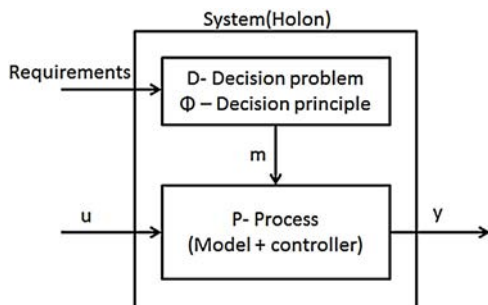


Fig. 3. The goal seeking system for HMS.

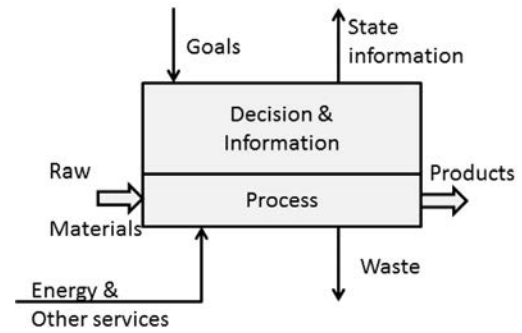


Fig. 4. Production unit.

- *Unit element*: A process unit element represents a physico-chemical processing task;
- *Header element*: A header element represents the logistics of materials or services within a specific segment of the overall process network;
- *Service element*: A service supplier element represents a custodian responsible for allocating a service to customer process elements that use this service in their local tasks;
- *Product element*: A product element represents the production requirements of a specific customer order in the form of a product recipe (specifying the sequence of processing tasks to be used or allowed) and other requirements such as quality, quantity and throughput of the product demand.

For each work, we have identified a strong point that will be the basis of the model presented in the next section. The works in [9,10] provides a control framework; the structure includes the process, the decision algorithm and application of this decision. Furthermore, all information handled by the algorithm is obtained via a data acquisition system directly on the system. Finally, the works in [8] based all his work on the principle of production tasks. This concept is similar to the notion of service that can be found in computer science. In our work, we propose to use the concept of service in order to define the product lines and capabilities of resources.

HCS are conventionally modeled by switching patterns using the whole system instead of atomic resources. Therefore, the reconfiguration process is complex because it must take into account the system as a whole, making the HCS inflexible and more susceptible to uncertainties (less reactive). Thus, these holonic architectures are implemented to search a more flexible HCS, although in a less generic way. An objective of this paper is to propose an alternative holonic control model based on a very generic holonic manufacturing architecture.

3. Holonic hybrid control model

The application of holonic paradigm to hybrid control systems systematically starts with the same interrogation: what can be considered as the product? Indeed, the discrete-event oriented approach is focused on the products and on the treatments the different machines and humans operate [17–19]. The nature of the products handled by hybrid control systems does not enable the application of these concepts. Furthermore, the interconnectivity problem is different: either it is often made with batches of products, which can be seen as discrete products enabling activity of the batch container, or it is a continuous piping network, and it is a set of sensors which retrieve the characteristics of the product. Focusing on the sole hybrid behavior, it is therefore the machines

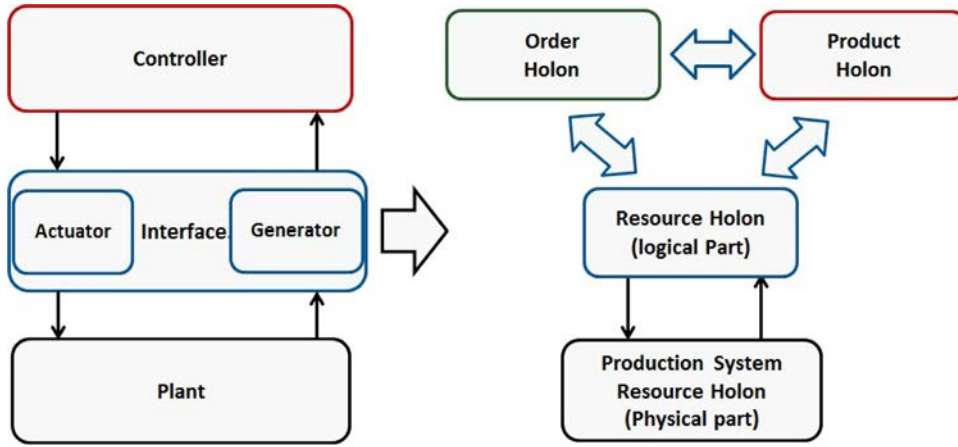


Fig. 5. (a) Hybrid control systems and (b) holonic hybrid control system.

which are devoted to be parts of decision making, and not really the products themselves.

This section introduces the holonic architecture application into HCS, as is showed in Fig. 5. The parts which composed the HCS, the controller, the interface and the plant, are reformulated in the proposed holonic hybrid control model (H²CM). The controller is represented by the product and order holons, the interface is represented by the logical part of the resource holons, and the production system is represented by the physical part of the resource holons. Thus, the product holon, resource holon and order holon are defined and a conceptual model of orchestration is proposed. Then, the main negotiation mechanism is detailed in order to illustrate the choreography of the holons. By the end, a simple supervision system is described.

3.1. Holons definition

The H²CM is based on the three basic holons of PROSA. As previously introduced, the staff holons are not mentioned on purpose, as their role is difficult to express in a generic way. Furthermore, basic holons objectives are globally the same than in a discrete configuration, so they can be considered as a first approximation and they are not presented in detail in this study. Fig. 6 introduces the basic building blocks diagram of H²CM. Two

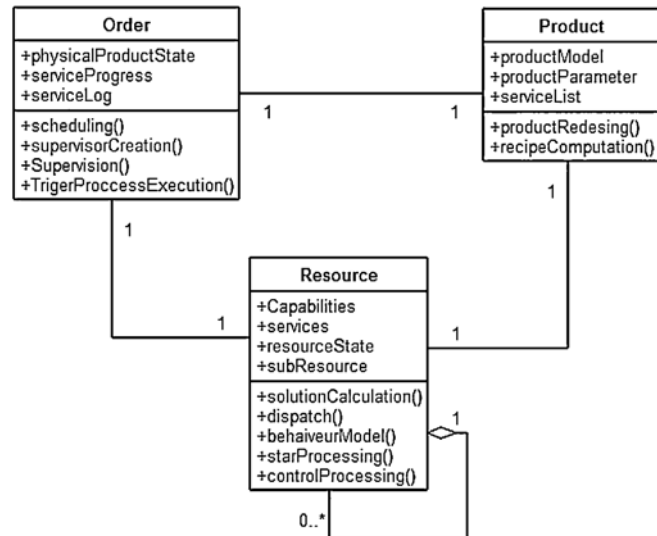


Fig. 6. Basic building blocks diagram of holonic hybrid control system.

main differences with the discrete event modeling expressed in Fig. 2 appear:

1. Each resource is granted with an order and a product along its life. The order holon is in charge of the monitoring of the resource whereas the product holon is in charge of the recipe to be applied on the actual product. The content and objectives of the order and product holons are constantly evolving, but the structure remains permanently the same;
2. A recursivity link is present on the resource holon. Indeed, each compound resource can be fractally decomposed into one or several holarchies, comprising one or several resources and their associated order and product holons. Unlike the previous structure, the aggregation relations created here can be changed along the working of the system; holarchies can be created and destroyed online.

The order holon is globally responsible for the planning of the system. It is very representative of the dual working of every holons: one part of the holon (denoted real time RT in the following) is dedicated to the system, while the other part (look ahead LA in the following) is dedicated to the negotiation mechanisms in look-ahead mode in order to determine the future planning and holarchies of the system. RT order is meant to monitor the inputs and outputs of the system via sensors in order to fit the requirements expressed by the RT product. It is also responsible for the application of the planning of the resource resulting from the previous negotiations. LA order is an actor of the negotiations, as defined in the next section.

The product holon has one main difference with the discrete one. Traditionally, this holon is dedicated to store and communicate the recipes of the products to manufacture and currently this holon uses a NEU protocol [20,21] to improve recipe information update. In the case of HCS, the recipes are deduced from a Master Recipe. In H²CM this Master Recipe can be defined as a generic recipe, i.e. a sequential list of operations to be applied to the product to obtain a final product from raw materials (BOM – Bill of Materials), from which the actual recipe can be derived according to the conditions of the system. For example, according to the quality of the raw oil, it can be necessary to add or subtract filtration operations from the first steps of the recipe. In HCS context, a service-oriented specification, as proposed in [22], is well suited for the product specification. The distinction made in this article, with respect to the definition used in [22], is that the parameters and variables of the service can be continuous or discrete. Although the NEU protocol [20] is not used, it could be considered for future work. The whole recipe is thus distributed

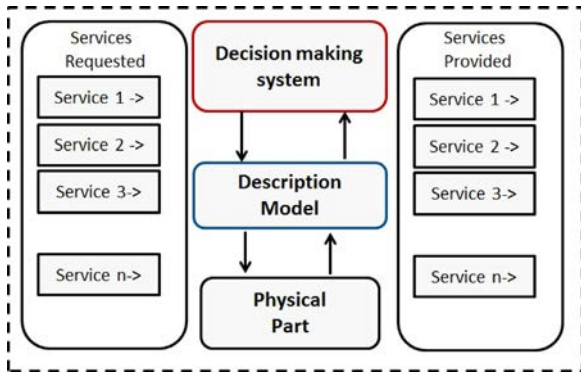


Fig. 7. Resource holon communication model.

along the whole holarchy, and a product holon is specified by 3 elements:

- Quality indicators of the raw material;
- Quality indicators of the final product;
- Definition of the service to be executed by the resource.

By nature, HCS are large systems, constituted of many components. Therefore, a lot of resource holons are necessary to control the system. As Fig. 6 defined a recursivity of the resources, it is necessary to determine the smallest size of the resource holons. The smallest resource holon to be defined are called atomic resources in the following and can be expressed as “the maximal aggregation of elements whose system of differential equations can be inverted in a short delay relatively to the dynamics of the system”. Considering compound resources, the negotiation mechanism is meant to determine the best solution recursively interrogating the aggregated resources until atomic ones. LA resources are part of the negotiation mechanisms. Their function is to evaluate and transmit to the order holon the best possible variables values to obtain the desired function and services. Negotiation mechanisms uses a communication model, see Fig. 7, where the negotiation language of resource is based in services parameterized, resources request services and provides services, as proposed in [22]. RT resource holons have the responsibility for the devices online control, i.e. the role of controllers of the system (Fig. 1).

Other resource holon specificity is its structure. Classically, it is composed by a physical part and a logical part, see Fig. 8. The physical part is represented by the shop floor. The logical part of the resource holon is an abstraction of the physical part and

contains the conversion models from continuous states to discrete states and vice-versa. The models used are hybrid models that change their state using threshold levels of continuous variables. Several models can coexist in the resource holon and represent features such as failure detection model, the desired behavior model and models that are inherent in holonic paradigm as a negotiation model for example.

3.2. Negotiation mechanism (LA function)

As stated before, the H²CM is very similar to the discrete oriented one. However, the functionalities of each holon are slightly different, as some tasks are adapted due to the nature of the product. Fig. 9 shows a comparison between the architecture expressed in [15] and the model suggested here. A first difference can be denoted as it is a service-oriented architecture. Therefore, the tasks are replaced by the notion of service, which is quite similar at this level of modeling. Considering the orders, the notion of deadlock is suppressed, as this is not meant to happen in a continuous process. The main differences can be spotted in the resource functions, as the role of resources is more important in this field.

Fig. 10 shows a sequence diagram of the negotiation when a new order is placed at a level $N + 1$, and how it interacts with subresources of level N . This sequence is executed each time a new negotiation is triggered (LA function), i.e. when an effective new order is meant to enter the system and has to be inserted in the planning of the resources, or when an order needs to be reevaluated in case of impossibility for the resources to executed the planning at the previously given cost.

Sequence n° 12 (denoted Dispatch()) is the basis for fractal behavior of resources. In the context of compound resources, this sequence repeats the set of sequences n7–17, until reaching atomic resources level. Therefore, Fig. 10 can be seen as two Dispatch() functions imbricated, from level $N + 1$ to level N . Sequence n13 allows the resources to compute the best solution among the different propositions offered by the subresources. Sequence n14 is generally a resolution of differential equations in an atomic resource. Step n3 consists in computing the recipe of the product feasible on the system with the given resources capabilities, without knowledge of the planning of the resources.

3.2.1. Systems supervision

The supervision is performed by an order holon (RT). As the proposed structure is meant to be fractal, each order holon has an associated resource holon. Consequently, the supervision is a direct supervision between orders and resources, which greatly simplifies the task of designing the supervisor in comparison to

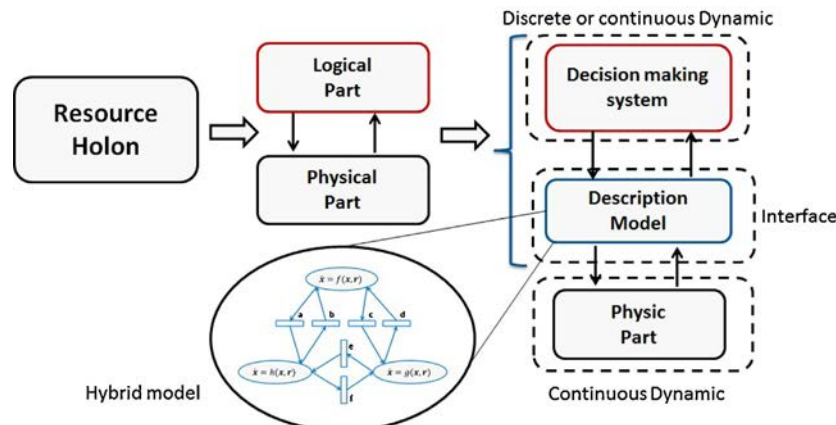


Fig. 8. Resource holon structure.

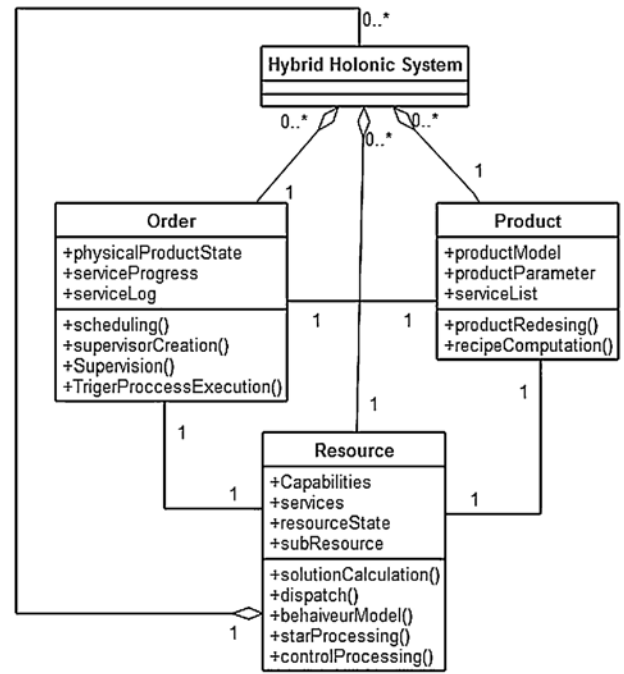
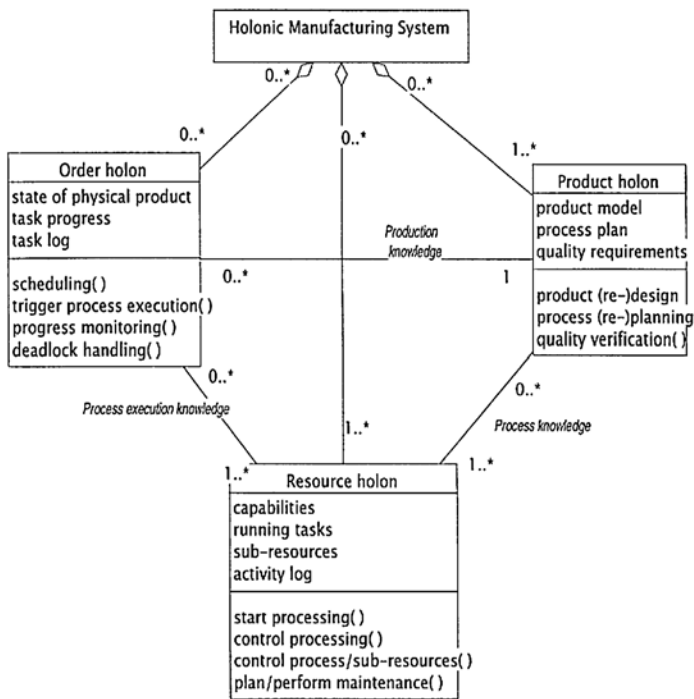


Fig. 9. Comparison between discrete [15] (left) and hybrid (right) holonic systems architectures.

classical supervisor designed for the whole system. The supervisor's structure is a fixed and simple one, focused on accomplishing the planning following the stages of the associated product (recipe computation). Another supervisor task is to verify the resource holon states. Hence, two scenarios are presented in the following. In the first scenario, the order holon is associated with an atomic resource, therefore its state only depends on the atomic resource state. In a second scenario, the order holon is associated with a compound resource. There, the order state is still related to the state of the resource it is associated with, but this resource state depends on the states of the resources it contains.

Resource states defined in this paper are "resource in failure", "resource in maintenance", "operative resource" and "degraded operative resource". The state of the compound resource is determined by the concatenation states of the resources they contain, i.e. if resource states are represented by automata, compound resource states are represented by automata concatenation. To illustrate this model an example is used with a resource A that is composed by two redundant resources B and C, resource A states are defined by the following Fig. 11. This case is representative of resources placed in parallel. The case of resources placed in series is trivial, as the state of the whole series is the state of the most limitative resource.

4. Study case

A study case of industrial electricity generation process was chosen, specifically a combined cycle plant (CCP) [23,24], which is a hybrid system where the turbine behavior represents the continuous dynamics and the turbine planning represents the discrete dynamics. A holonic control model and an emulation of the physical CCP were developed in Java and a benchmark from the literature was chosen in order to validate the efficiency of the approach.

This section begins with the CCP description, followed by the models description used in the representation of the turbines in the emulation of the CCP. In the third part, the holonic control

model is described, followed by the experimentation definition. Finally results and discussion are presented.

4.1. Combined cycle plant description

A CCP is a thermodynamic process, which aims to produce electrical energy through two different cycles [25]. The first one is an open cycle with air and fuel inlets. The gas turbine is an internal combustion engine, with a combustion chamber to be supplied with fuel (diesel or gas) and air. This combustion generates gas with sufficient capacity to drive a turbine coupled to a synchronous generator. During this cycle, almost 65% of energy is wasted, mainly dissipated in heat, which induces a very high cost of energy (in /kWh).

The arrangement of the combined cycle gives an additional 50% of the electrical energy and 60% conversion efficiency. The basic configuration of a combined cycle is shown in Fig. 12.

4.2. Study case model

The objective of this section is to describe the models used in the emulation of CCP. The models were retrieved from literature: gas turbines model is presented in details in [26] and denoted Heavy Duty Gas Turbine (HDGT) in the following, steam turbine model is presented in details in [24].

4.2.1. Gas turbine model

HDGT are specially designed gas turbines for power generation which are specified by their long life and higher availability compared to other types of gas turbines [26]. HDGTs are composed of three major components: multistage axial flow compressors, can-annular combustors and axial flow turbines. Air with atmospheric conditions is drawn to the compressor after passing air filters at the entrance. The multi-stage compressor increases speed, pressure and temperature of air before it reaches the combustor and inlet to the high pressure turbine parts.

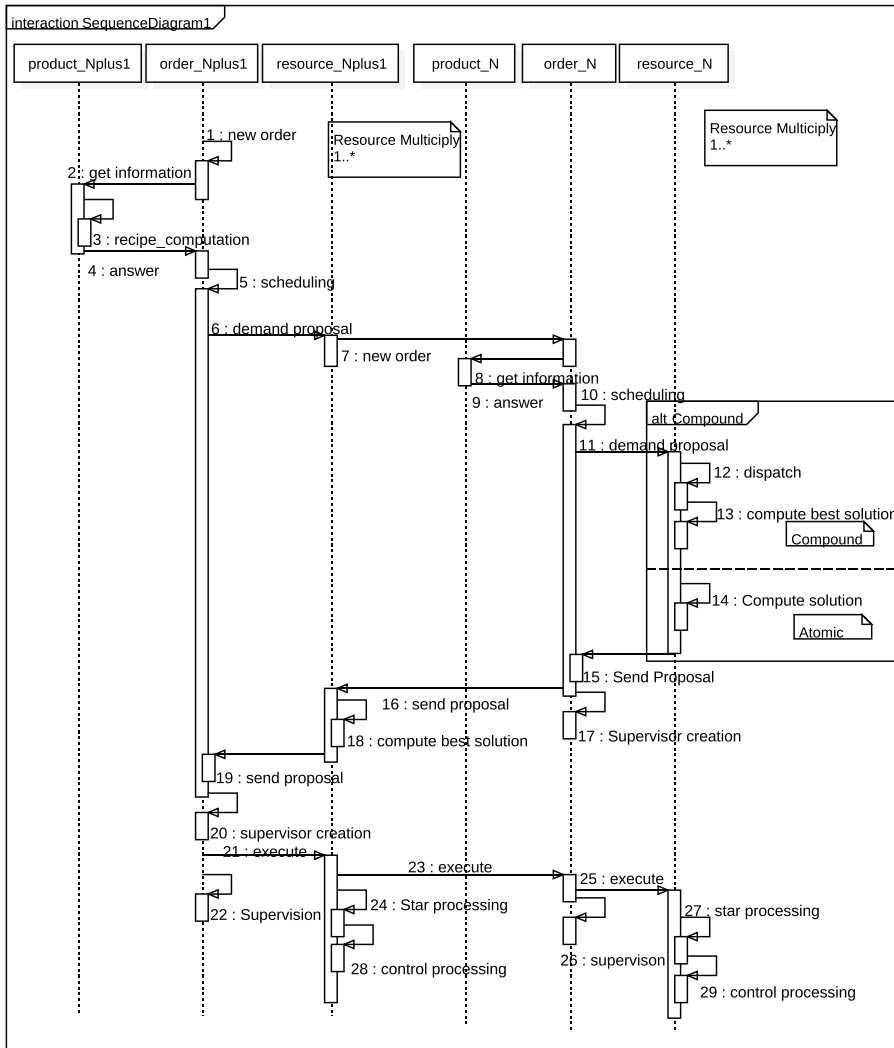
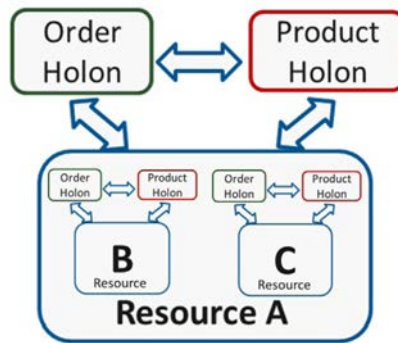


Fig. 10. Planning negotiation sequence diagram.



Resource B	Resource C	Resource A
Operative	Operative	Operative
Operative	Maintenance	Operative-degraded
Operative	Failure	Operative-degraded
Operative	Operative-degraded	Operative-degraded
Maintenance	Maintenance	Maintenance
Maintenance	Failure	Failure
Maintenance	Operative-degraded	Operative-degraded
Failure	Failure	Failure
Failure	Operative-degraded	Operative-degraded
Operative-degraded	Operative-degraded	Operative-degraded

Fig. 11. States expression of a resource composed of redundant subresources.

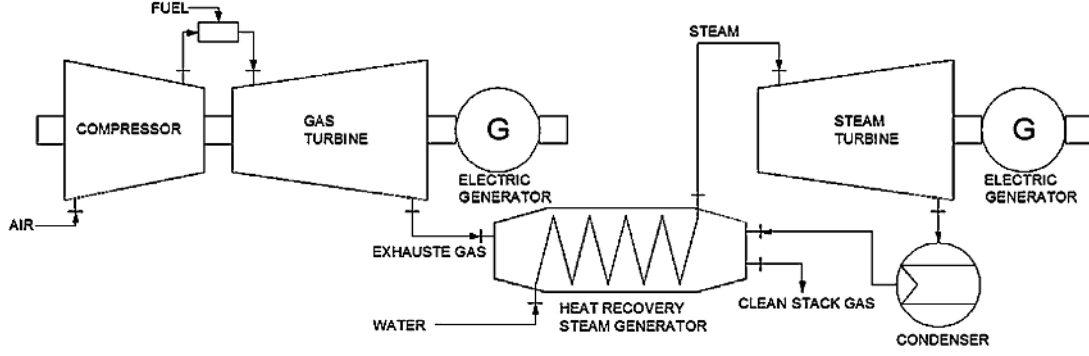


Fig. 12. Combined cycle plant basic configuration.

Each compressor stage comprises a row of rotor blades and stator vanes. The compressed air with high pressure and temperature follows its way to the combustor. The combustor is in essence a heater in which fuel is burnt to increase the temperature at a more or less constant pressure. Roughly one third of the compressor discharge air is mixed with the fuel to be burnt. The remaining volume of air is mixed with the products of the combustion to become the turbine inlet flow, which is thus already at turbine inlet temperature. The flow is then expanded in turbine stages, which drive compressor and generator. Finally, the flow is guided through the exhaust duct to a second environment which can be a surrounding ambient conditions or a heat recovery steam generator (HRSG) in combined cycle plants (CCP).

In Fig. 13, the HDGT model of [26] is shown. The turbine behavior is reflected by two quantities: the output torque T_{mech} and the exhaust gas temperature $T_{exhaust}$.

After some calculations, the equations can be expressed as the power block shown in Fig. 13 as follows:

$$P_{Gpu} = A + Bm_{fpu} \quad (1)$$

at nominal speed () where A and B are the coefficients of output torque in Fig. 13, m_n and m_{fn} are the air and fuel nominal flow rates. P_{Gpu} is the per unit output power, which is equal to the $p.u.$ torque. The turbine nominal power is the base of $p.u.$ At nominal speed, the exhaust temperature can be calculated using the following

equation:

$$T_E = T_R - D(1 - m_{fpu}) \quad (2)$$

at nominal speed (), where D is the coefficient of the exhaust temperature block in Fig. 13, T_R is the nominal exhaust temperature of the HDGT and T_E is the gas turbine exhaust temperature.

To extract the turbine parameters, a typical operational condition is selected to derive the turbine and compressor efficiencies and then all above parameters are computed with available data out of unit operation. These parameters [27] are given for a 100 MW turbine in Table 2. The turbine is a single shaft HDGT with natural gas as primary operating fuel.

Using the model of Fig. 13 and parameters of Table 2, a HDGT model was programmed using Matlab-Simulink. This model was useless for the emulation, but is used for the validation of the discretized emulation model. Indeed, Simulink block diagrams and some characteristic response can be found in literature. The idea is therefore to compare the results obtained with the Java model to this Simulink model in order to validate the discretization.

To make and simulate a java model, Eqs. (1) and (2) were explicitly discretized in time, following the control structure of Fig. 13 and parameters of Table 2. For the Java model validation, a step function was used to carry up the turbine from 0% to 50% and 100% load. Only little differences exist between models, and only

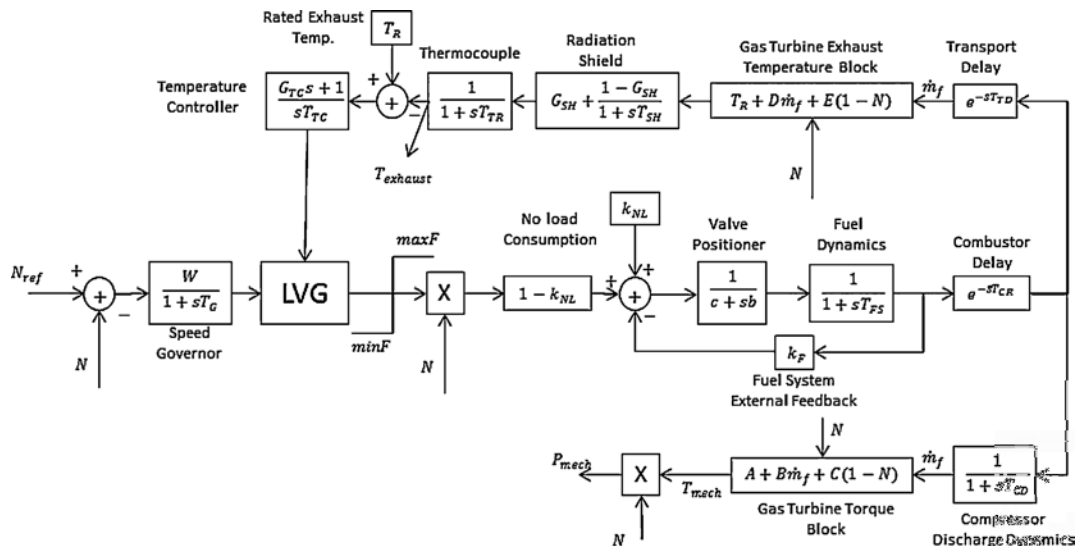


Fig. 13. HDGT model.

Table 2
Parameters of HDGT model.

Parameter	Symbol	Value
Speed governor gain	W	25
Speed governor time constant	TG	0.05
Fuel demand signal max limit	$maxF$	1.5
Fuel demand min limit	$minF$	-0.13
No load fuel consumption	KNL	0.24
Valve positioner time constant (s)	B	0.04
Fuel system time constant (s)	Tfs	0.26
Fuel systems external feedback loop gain	KF	0
Delay of combustion system (s)	TCR	0.005
Transport delay of turbine and exhaust system (s)	TTD	0.04
Compressor discharge lag time constant (s)	TCD	0.16
Gas turbine torque block parameter	A	-0.158
Gas turbine torque block parameter	B	1.158
Gas turbine torque block parameter	C	0.5
Gas turbine exhaust temperature block parameter ($^{\circ}C$)	D	413
Gas turbine exhaust temperature block parameter ($^{\circ}C$)	E	313
Radiation shield parameter	GSH	0.85
Radiation shield time constant (s)	TSH	12.2
Thermocouple time constant (s)	TTR	1.7
Rate exhaust temperature ($^{\circ}C$)	TR	522
Temperature controller parameter	GTC	3.3
Temperature controller integration constant ($^{\circ}C$)	TTC	250

during transition phases. These differences are relatively short (order of a minute) compared to the system dynamics (hour), and these differences are not taken into account in the processed results, so these were considered as meaningless.

4.2.2. Steam turbine model

The heat recovery steam generator (HRSG) reacts to the changes in gas turbine exhaust flow W , and exhaust temperature T_E . The steam turbine mechanical power response follows changes in gas turbine exhaust energy with lags for the high pressure contribution and lags for the low pressure contribution.

It is very likely that the total contribution to mechanical power from the two pressure boilers can be approximated with a two time constants model. The gain between gas turbine exhaust energy and steam turbine output will in general be a nonlinear function which can be derived from steady state measurements through the load range, or from design heat balance calculations for rated and partial load conditions. The chosen HRSG model and steam turbine model are thus the combined simplified model presented in [24], which can be represented as the schema of Fig. 14. This prevents us from decoupling the HSRG and the steam turbines, which is a small approximation for the control, but the simplification brought by this model is worth the approximation.

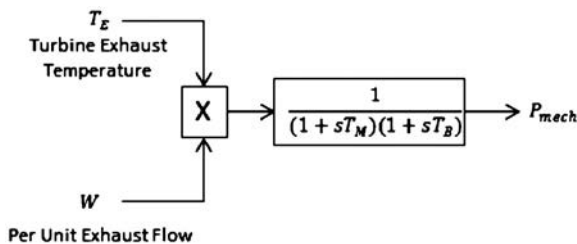


Fig. 14. Simplified steam turbine model.

4.2.3. Java modeling

CCP was emulated using Java. Eight classes were created over the main class: GasTurbine, SteamTurbine, GasTurbineSim, SteamTurbineSim, Resource, Product, Order and service (Fig. 15).

HDGT and steam turbine models were programmed in GasTurbine and SteamTurbine classes, a “Simulation” method was created and it contains all gas and steam turbines equation, these classes come to represent the turbines physical parts. The GasTurbineSim and SteamTurbineSim classes are an abstraction of the GasTurbine and SteamTurbine classes and they represent the logical part of the gas and steam turbines. Inside these classes, the “descriptionmodel” method contains a graph representing all possible states of the turbines.

Resource class is the class where all the capabilities of resources such as services, capabilities, and cost were defined. It also contains “dispatch” and “solutionCalculation” methods whose are parts of the negotiation mechanisms to determine the best configuration of production.

Finally, there is the “Service” class. In this class is defined the structure of the provided and required services. Basically the service defined in the CCP emulation is the electricity generation with parameters specification such as frequency and type of fuel used.

4.3. Holonic hybrid control model for CCP

The first step to implement the H²CM in a CCP is to define all the holons in the system. This definition is done following the previously proposed fractal structure, where the maximal aggregation is represented by a gas and a steam turbine and defined as atomic holons. A 4×2 arrangement (four gas turbines for two steam turbines) has been chosen. The arrangement and connection of the units represented in the 2×1 configuration forms a compound holon. Thus, the system has two compound holons with the same structure that are called resource holon 2 and 3, which in turn form a third holon called resource holon 1, as shown in Fig. 16. The product holon is defined by electric generation service from each turbine, service that can be parameterized by different values like frequency, fuel type, etc. There is a distinction between gas turbines and steam turbines products, gas turbine has the generation energy and heat generation as products and steam turbine only has a power generation as product and needs the product of heat generation for its operation.

During the planning process, holons negotiate with each other as shown in the sequence diagram of Fig. 17. Each compound resource parameterizes a new suborder with its needs (sequence n7), the atomic holon according to its capacity and availability computes a solution (sequence n12) and sends a proposal (sequence n14). The compound holon receives all the propositions and calculates the best solution (sequence n16), thus generating the planning and the holarchy that will execute the planning.

The computation of the solution of compound holon (sequence n16) is performed following the algorithm of Fig. 18. The terminology used in the algorithm is the following

- $D_r(t)$: resource demand during time t ;
- R_i : resource i inside compound resource R ;
- C_{R_i} : resource i capacity
- N : number of resources available inside R ;
- $C_R = \sum_{i=1}^N C_{R_i}$;
- $L_{R_i}(t)$: resource i load;
- G_i : gas turbine i ;
- $L_{G-i}(t)$: gas turbine i load;
- S_j : steam turbine j ;
- $L_{S_j}(t)$: steam turbine j Load;

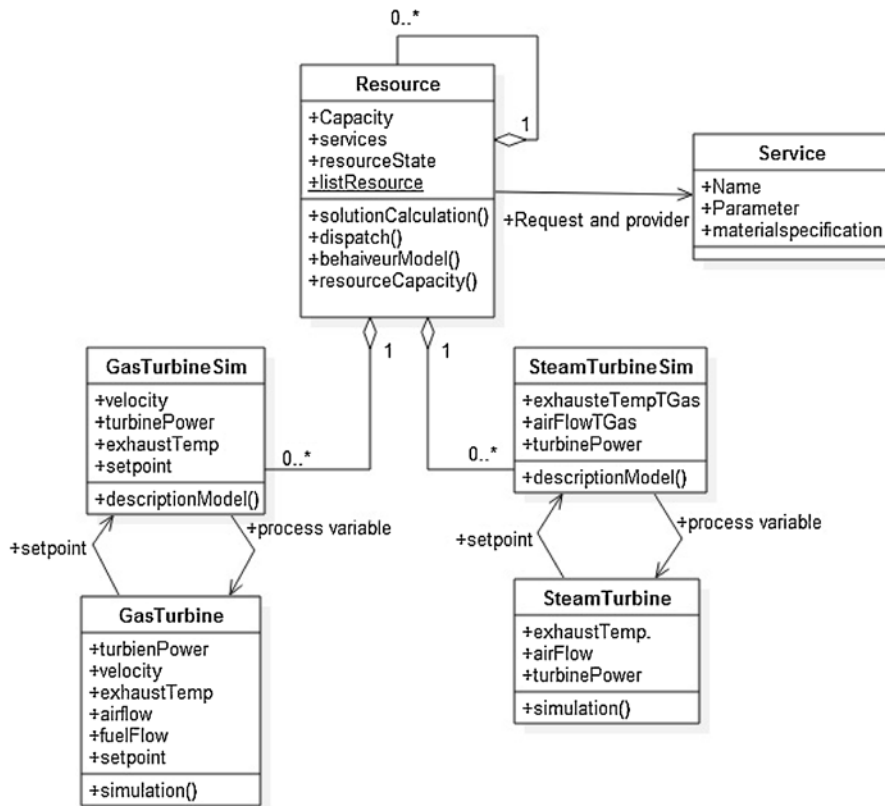


Fig. 15. Combined cycle plants class diagram.

- $\$R$: resource R Cost;
- t : planning time (in hours);
- t_1 : initial planning time period;
- t_2 : final planning time period.

Each time the algorithm is executed, it needs to be parameterized with the required energy demand, the period and the resources to plan. It is basically divided into two cases, the first case where the compound holon is formed by compounds holons, then planning resources is a function of three variables: capacity, availability and cost of the resource. First, the cheapest resource is chosen and the available capacity is allocated. The second case

occurs when the resource is composed by atomic resources and the required load is distributed evenly between the available resources considering their minimum capacity. The goal of using every atomic resource of the compound holon is to increase the reactivity of the system: in case of failure, atomic resources are already available and started, so it is only needed to adjust their set point in time to compensate the failure. Otherwise, needing to start from scratch a turbine would not allow to react quickly enough to failures.

Finally, the supervisor used in implementation of H^2CM checks the status of the resource. At the time the resource changes state (for example when resource changes from operative to degraded), the supervisor detects the new status (in this case a failure) and executes a production re-planning parameterized which the period and the resource to re-plan.

4.4. Definition of experiments

In order to experimentally verify the results and impacts of the proposed model, an experimentation is introduced with power plants dispatching data on a 24-h period, and two turbine failures. It is important to clarify that the purpose is not to evaluate the optimization of plans elaborated by HCM and production execution, but check and evaluate compliance with the attributes of holonic model as proposed, good performance on HCS, failure compensation and flexibility of system. The characteristics of capacity and costs of each turbine used in the study case are shown in Table 3.

In order to test the hypothesis of a properly functioning of H^2CM , a method is defined to check the planning accomplishment, resolution of resource degradation failure and resolution of resource stopping failures. Thus, the following experimental methodology is defined:

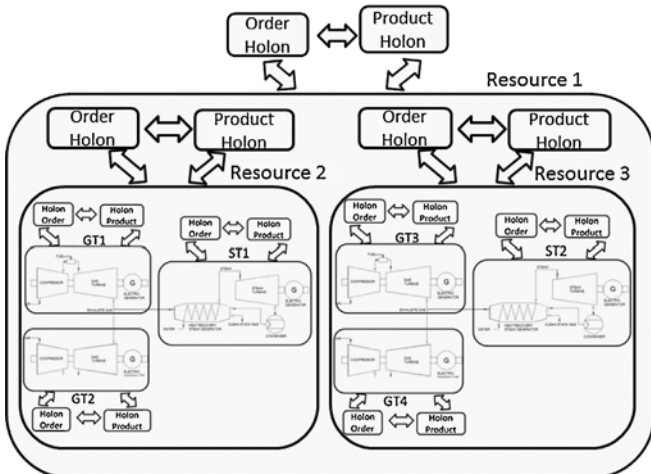


Fig. 16. CCP holonic system (fractal system).

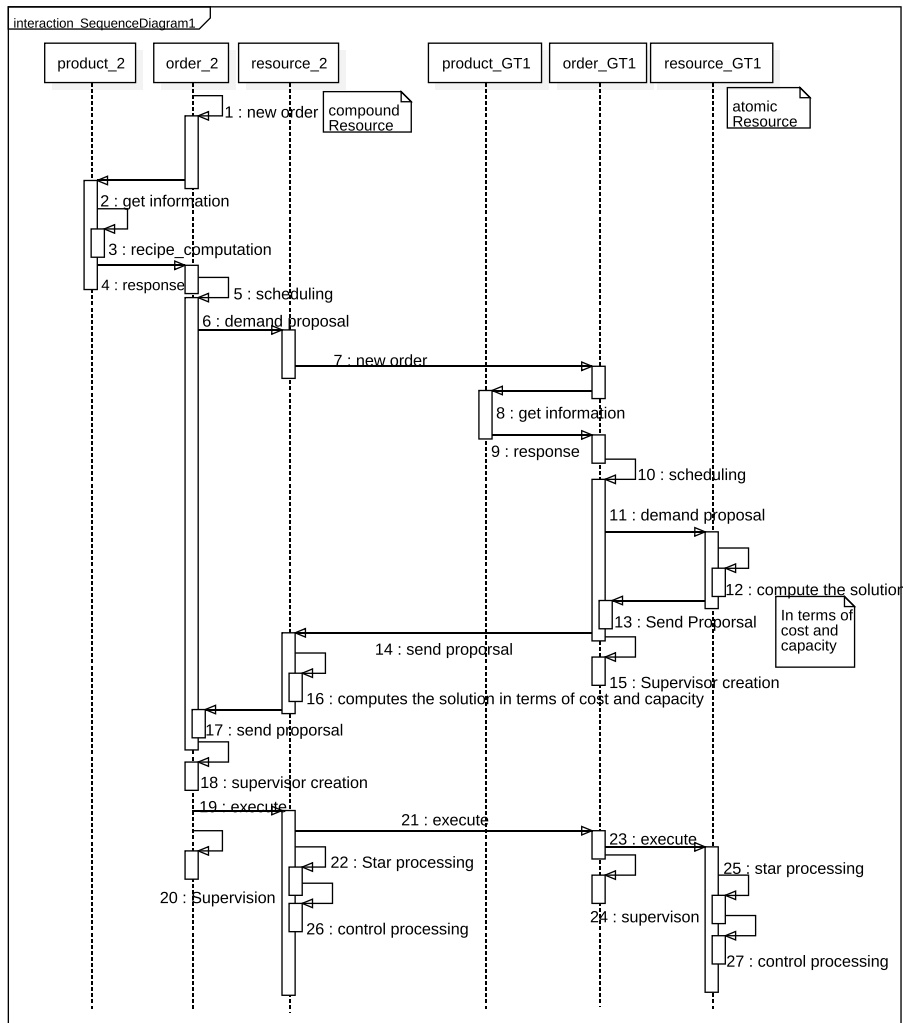


Fig. 17. CCP planning sequence diagram.

- Define a mission over a period of 24 h for each experiment;
- Plan production and form the holarchies;
- Generate failures or degrade the capacity of resources, Table 4;
- Re-plan the production.

4.5. Results and discussion

4.5.1. Planning

The first result showed refers to the demand planning and creation of holarchies. In Table 5, the total energy demand of the system is observed in r1, it is also observed the load designation by hour to each resource and how these resources form the holarchies. Examples of holarchies formed are during periods 5–11, the resources involved are resource r1, r2, GT1, GT2 and ST1, while during the periods 12–21 the holarchy consists of all resources. This planning results from the instantiation of the algorithm showed in Fig. 18, the compound resource r2 has lower production costs that the compound resource r3, therefore to r2 is assigned a greater load. The allocation of the load on the atomic resources inside r2 is made equally among all available resources.

4.5.2. Flexibility

Failures are intentionally introduced into the system to observe their behavior. The first system failure is a ST1 resource

degradation of capacity from 100 MW to 80 MW during 20 min (period 11). The reaction of the system is to compensate the loss of production capacity using r2 atomic resources and assigns the missing load to GT2. Given this failure, the holon r2 has the ability to compensate the loss, demonstrating autonomy of holon. As the failure lasts only 20 min, the generation capacity of the turbine ST1 is restored for the next hour.

In the second failure, a complete stop of production of GT2 occurs (period19). This failure might be considered as a major failure and the holon r2 do not have enough generating capacity to compensate the loss of production. Therefore, r2 requires collaboration from holon r3 through holon r1. The system increases the load designation to r3 to compensate the capacity loss of r2 (collaboration). These system reactions can be observed in Table 6.

The planning, production execution and difference between them is shown in Fig. 19, noting that the execution follows all the time the planning, with short periods of overproduction in large changes in demand as set point turbines are adjusted before the time planning of the effective increase of the demand. It is also possible to observe the system reaction to failure during periods 11 and 19, as electricity generation falls down and the system automatically compensates, but with a delay due to the dynamic of the turbines.

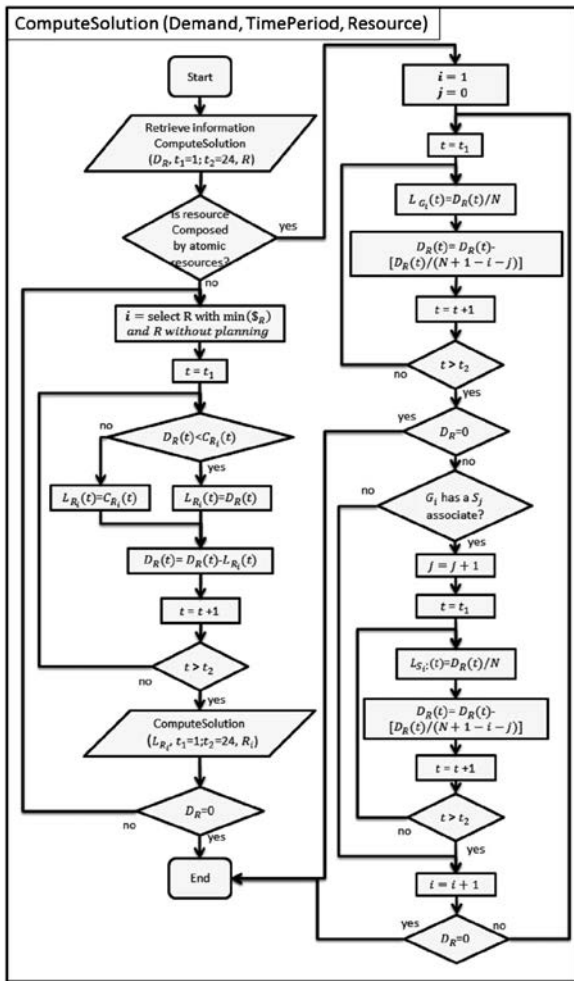


Fig. 18. Compute best solution algorithm.

Table 3
Resource capacity.

Holon	Cost (USD/MW)	Maximum capacity (MW)	Minimum capacity (MW)
Gas turbine 1 (GT1)	1000	100	15
Gas turbine 2 (GT2)	1100	100	15
Gas turbine 3 (GT3)	1200	100	15
Gas turbine 4 (GT4)	1300	100	15
Steam turbine 1 (ST1)	400	100	10
Steam turbine 2 (ST2)	500	100	10

Table 4
Failure type in experimentation.

Failure	Period	Resource in failure	Type of failure	Description
1	11	ST1	Degradation	from 100% to 80% of capacity
2	19	GT2	Failure	from 100% to 0% of capacity

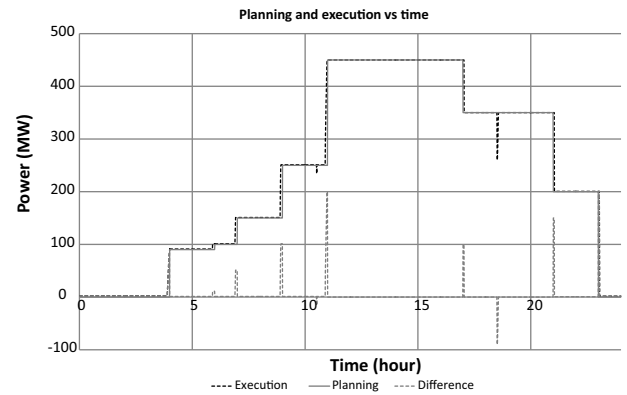


Fig. 19. Production planning and execution.

Table 5
Production planning.

Period	r1	r2	r3	GT1	GT2	ST1	GT3	GT4	ST2
1.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5.0	90.00	90.00	0.00	30.00	30.00	30.00	0.00	0.00	0.00
6.0	90.00	90.00	0.00	30.00	30.00	30.00	0.00	0.00	0.00
7.0	100.00	100.00	0.00	33.33	33.33	33.33	0.00	0.00	0.00
8.0	150.00	150.00	0.00	50.00	50.00	50.00	0.00	0.00	0.00
9.0	150.00	150.00	0.00	50.00	50.00	50.00	0.00	0.00	0.00
10.0	250.00	250.00	0.00	83.33	83.33	83.33	0.00	0.00	0.00
11.0	250.00	250.00	0.00	83.33	83.33	83.33	0.00	0.00	0.00
12.0	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
13.00	50.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
14.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
15.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
16.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
17.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
18.00	350.00	300.00	50.00	100.00	100.00	100.00	16.67	16.67	16.67
19.00	350.00	300.00	50.00	100.00	100.00	100.00	16.67	16.67	16.67
20.00	350.00	300.00	50.00	100.00	100.00	100.00	16.67	16.67	16.67
21.00	350.00	300.00	50.00	100.00	100.00	100.00	16.67	16.67	16.67
22.00	200.00	200.00	0.00	66.67	66.67	66.67	0.00	0.00	0.00
23.00	200.00	200.00	0.00	66.67	66.67	66.67	0.00	0.00	0.00
24.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 6
Systems re-planning from failure.

Period	r1	r2	r3	GT1	GT2	ST1	GT3	GT4	ST2
1.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5.0	90.00	90.00	0.00	30.00	30.00	30.00	0.00	0.00	0.00
6.0	90.00	90.00	0.00	30.00	30.00	30.00	0.00	0.00	0.00
7.0	100.00	100.00	0.00	33.33	33.33	33.33	0.00	0.00	0.00
8.0	150.00	150.00	0.00	50.00	50.00	50.00	0.00	0.00	0.00
9.0	150.00	150.00	0.00	50.00	50.00	50.00	0.00	0.00	0.00
10.0	250.00	250.00	0.00	83.33	83.33	83.33	0.00	0.00	0.00
11.0	250.00	250.00	0.00	83.33	86.67	80.00	0.00	0.00	0.00
12.0	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
13.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
14.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
15.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
16.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
17.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
18.00	350.00	300.00	50.00	100.00	100.00	100.00	16.67	16.67	16.67
19.00	350.00	200.00	150.00	100.00	0.00	100.00	50.00	50.00	50.00
20.00	350.00	200.00	150.00	100.00	0.00	100.00	50.00	50.00	50.00
21.00	350.00	200.00	150.00	100.00	0.00	100.00	50.00	50.00	50.00
22.00	200.00	200.00	0.00	100.00	0.00	100.00	0.00	0.00	0.00
23.00	200.00	200.00	0.00	100.00	0.00	100.00	0.00	0.00	0.00
24.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

5. Conclusions

The work proposed in this paper concerns the implementation of holonic architecture on HCS. The presented model tends to confirm the good fit of the classical reference architecture PROSA to this class of systems. The holonic architecture also provides its intrinsic simplicity, simplest management models behavior and flexibility, with the possibility to reconfigure resources. This work introduces a performance evaluation of holonic control on a dynamic HCS. As a first result, the production planning is obtained using the negotiation of resources. Although no optimal method is used to obtain this result, it is quite possible to implement any optimal method inside the holons for a best result. The second tangible evaluation is the response of the system facing failures such as resource degradation and resource full stop. H²CM achieves to compensate the loss of production in a very short reconfiguration delay.

Another feature to evaluate is the flexibility of H²CM to be reconfigured (remove or add resources). The experimental plant is modified making it bigger and the H²CM continuous working properly, the calculation time of planning are compared and obviously increases as the size of the plant increases, but remains low times from the process control point of view. With classical HCS model, modifications of the plant imply modifications of the control system, which is very time-consuming and decreases the systems flexibility. With H²CM, removing or adding resources does not alter the control structure, so no additional time is required to redesign the system, which is the key for systems flexibility.

Future works will therefore be oriented to find a new algorithm able to find optimal solution, study how to H²CM implementation affects the stability of HCS, and propose algorithm based on other holonic reference architectures such as ADACOR or HCBA to evaluate the simplicity of implementation. Another future work will be based in the inclusion of staff holon in the H²CM architecture, detailing the mechanism used by the model to switch staff holon status between idle state to active state. Once the staff holon is active, the architecture changes to a hierarchical structure and the system would have a behavior similar to classic

control systems. Finally, this paper has studied a specific family of hybrid systems called hybrid control systems. However, the family of hybrid systems has many elements, so there are many other subfamilies such as switched control systems that can also be studied.

References

- [1] D. Andreu, J. Pascal, R. Valette, Interaction of discrete and continuous parts of a batch process control system, in: ADEDOPS Workshop, Imperial College, London, 1995.
- [2] H. Aytug, M.A. Lawley, K. McKay, S. Mohan, R. Uzsoy, Executing production schedules in the face of uncertainties: a review and some future directions, *Eur. J. Oper. Res.* 161 (1) (2005) 86–110.
- [3] S.V. Mehta, R.M. Uzsoy, Predictable scheduling of a job shop subject to breakdowns, *IEEE Trans. Robot. Autom.* 14 (3) (1998) 365–378.
- [4] R.F. Babiceanu, F.F. Chen, Development and applications of holonic manufacturing systems: a survey, *J. Intell. Manuf.* 17 (1) (2006) 111–131.
- [5] J. Barbosa, P. Leitão, E. Adam, D. Trentesaux, Self-organized holonic manufacturing systems combining adaptation and performance optimization, in: *Technol. Innov. Value Creat.*, Springer, 2012, pp. 163–170.
- [6] P.J. Antsaklis, Hybrid control systems: an introductory discussion to the special issue, *IEEE Trans. Autom. Control* 43 (4) (1998) 457–460.
- [7] X.D. Koutsoukos, P.J. Antsaklis, J. Stiver, M.D. Lemmon, et al., Supervisory control of hybrid systems, *Proc. IEEE* 88 (7) (2000) 1026–1049.
- [8] N. Chokshi, D. McFarlane, A distributed architecture for reconfigurable control of continuous process operations, *J. Intell. Manuf.* 19 (2) (2008) 215–232.
- [9] D.C. McFarlane, Holonic manufacturing systems in continuous processing: concepts and control requirements, in: *Proceedings of ASI*, vol. 95, Citeseer, 1995, pp. 273–282.
- [10] E. Chacón, I. Besembel, F. Narciso, J. Montilva, E. Colina, An integration architecture for the automation of a continuous production complex, *ISA Trans.* 41 (1) (2002) 95–113.
- [11] P. Leitão, F. Restivo, ADACOR: a holonic architecture for agile and adaptive manufacturing control, *Comput. Ind.* 57 (2) (2006) 121–130.
- [12] J. Chirn, D.C. McFarlane, A holonic component-based approach to reconfigurable manufacturing control architecture, in: *11th International Workshop on Database and Expert Systems Applications*, 2000, IEEE, 2000, pp. 219–223.
- [13] J. Barbosa, P. Leitão, E. Adam, D. Trentesaux, Dynamic self-organization in holonic multi-agent manufacturing systems: the ADACOR evolution, *Comput. Ind.* 66 (2015) 99–111.
- [14] C. Pach, T. Berger, T. Bonte, D. Trentesaux, ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling, *Comput. Ind.* 65 (4) (2014) 706–720.
- [15] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, P. Peeters, Reference architecture for holonic manufacturing systems: PROSA, *Comput. Ind.* 37 (3) (1998) 255–274.

- [16] P. Valckenaers, H. Van Brussel, J. Wyns, L. Bongaerts, P. Peeters, Designing holonic manufacturing systems, *Robot. Comput. Integr. Manuf.* 14 (5) (1998) 455–464.
- [17] Y. Sallez, Proposition of an analysis framework to describe the activeness of a product during its life cycle, in: *Service Orientation in Holonic and Multi-agent Manufacturing and Robotics*, Springer, 2014, pp. 257–270.
- [18] R. Pannequin, G. Morel, A. Thomas, The performance of product-driven manufacturing control: an emulation-based benchmarking study, *Comput. Ind.* 60 (3) (2009) 195–203.
- [19] X. Yang, P. Moore, S.K. Chong, Intelligent products: from lifecycle data acquisition to enabling product-related services, *Comput. Ind.* 60 (3) (2009) 184–194.
- [20] P. Valckenaers, P.A. De Mazière, Interacting holons in evolvable execution systems: the NEU Protocol, in: *Industrial Applications of Holonic and Multi-agent Systems*, Springer, 2015, pp. 120–129.
- [21] P. Valckenaers, H. Van Brussel, *Design for the Unexpected, From Holonic Manufacturing Systems towards a Humane Mechatronics Society*, first ed., Elsevier, 2015.
- [22] F. Gamboa Quintanilla, O. Cardin, P. Castagna, Product specification for flexible workflow orchestrations in service oriented holonic manufacturing systems, in: *Service Orientation in Holonic and Multi-agent Manufacturing and Robotics*, Springer, 2014, pp. 177–193.
- [23] W.I. Rowen, Simplified mathematical representations of single shaft gas turbines in mechanical drive service, in: *ASME 1992 International Gas Turbine and Aeroengine Congress and Exposition*, American Society of Mechanical Engineers, 1992, V005T15A001.
- [24] F. De Mello, D. Ahner, Dynamic models for combined cycle plants in power system studies, *IEEE Trans. Power Syst.* 9 (3) (1994).
- [25] G.D. Zapata Madrigal, *Propuesta para la planificación, programación, supervisión y control de la producción en procesos continuos desde la teoría del control supervisorio y el enfoque holónico* (Ph.D. thesis, Doctorado en Ciencias Aplicada), Universidad de Los Andes, 2011.
- [26] W.I. Rowen, Simplified mathematical representations of heavy-duty gas turbines, *J. Eng. Gas Turbines Power* 105 (4) (1983) 865–869.
- [27] M.R.B. Tavakoli, B. Vahidi, W. Gawlik, An educational guide to extract the parameters of heavy duty gas turbines model in dynamic studies based on operational data, *IEEE Trans. Power Syst.* 24 (3) (2009) 1366–1374.