

A holonic approach for manufacturing execution system design: an industrial application

Blanc Pascal, Castagna Pierre
IRCCyN – UMR CNRS 6597
1, rue la Noe BP 92101,
44321 Nantes Cedex 3, France
[blanc, castagna]@irccyn.ec-nantes.fr

Demongodin Isabel, Hennet Jean-Claude
LSIS – UMR CNRS 6168
Avenue Escadrille Normandie-Niemen,
13397 Marseille Cedex 20, France
[demongodin, hennet]@lsis.org

Abstract

Holonic Manufacturing Execution Systems (HMES) are structures organized as a loosely coupled network of communicating agents to answer in a flexible way to necessities of reactivity and profitability. In this paper, we propose a PROSA-based HMES and we illustrate our approach on a real industrial application. The working of this HMES relies on a predictive and reactive behavior to perform the control of production cells. Mechanisms are proposed to synchronize the execution of manufacturing orders, as to guarantee the availability of components and to set time-limit constraints on task scheduling. We also propose a typology of production cells, and specific heuristics to solve the tasks scheduling problem, taking into account each optimization model.

1. Introduction

Today's manufacturing systems are challenged by a demanding fast changing environment. Industrial firms have to implement control policies and adapt their manufacturing systems to maximize their productivity, considering also the manufacturing-customer responsiveness as a main issue [5]. The need for more reactivity, flexibility, and productivity leads to drastically increase the complexity of the (re)design of Manufacturing Execution Systems (MES).

Classical centralized and hierarchical approaches based on time or constraints aggregation show limitations to solve manufacturing control problems, particularly when the system faces numerous random events. Mass customization, small volume/high variety order management and maximization of the profitability of the firms are long-term challenges which require new approaches. These approaches support the use of cooperative and autonomous unit self-organized in an open structure to offer a very high operational and structural level of flexibility.

Holonic Manufacturing System (HMS) [4] is such an approach which combines the best features of

hierarchical and heterarchical organizational structures. This concept can preserve the stability of hierarchy while providing the dynamic flexibility of heterarchies and offers a suitable background to develop a distributed Manufacturing Execution System. To that end, the reference holonic architecture PROSA (Product-Resource-Order-Staff Architecture) is utilized [7]. PROSA is based on a typology of the manufacturing system elements: products, resources, orders and optional specialized decision-making centers (hierarchical and/or functional). It allows identifying and defining the roles and behaviors of the manufacturing control elements.

In this paper, we address the manufacturing control problem of a complex discrete manufacturing system. On the one hand, we propose mechanisms to synchronize the execution of manufacturing orders. These mechanisms guarantee the availability of components and enable to set time-limit constraints on task scheduling. In other words, it allows respecting precedence constraints. On the other hand, we propose a typology of Resource holons, based on characteristics of the tasks that they perform: supply, transformation, assembly or disassembly. For each type of Resource holon, specific heuristics are used to solve the tasks scheduling problem, taking into account the own model of each one. In the case of a disassembly holon, the scheduling problem may be associated to a disassembling pattern design problem. Thus an adaptation of the basic scheduling heuristic is proposed. These mechanisms and algorithms are adapted to a distributed manufacturing control system and may be implemented in a multi-agents system.

The contribution is organized as follows. In section 2, the context of the manufacturing control system is described; specific problems related to this control are addressed. The use of a PROSA-based holonic approach and the mechanism to perform the manufacturing control is presented in section 3. Finally, section 4 applies the proposed manufacturing control system to a real industrial application.

2. Manufacturing Control

A manufacturing system can be considered as a set of resources used to transform raw material into finished goods in order to satisfy a demand. In a systematic way, it is now usual to structure this system in three subsystems: physical, informational and decisional.

The transformation of raw material into finished goods is the result of successive transformations of single or multiple components into items by way of operations completed by the physical devices of the system. The informational subsystem gathers all the information existing in the manufacturing system, either static such as bills of material, manufacturing routing, standard operation duration, or dynamic such as current state of resource, location of an item. Finally, the decisional subsystem performs the management of the working of the whole manufacturing system. This one is in charge of fixing and putting into practice tactics to produce with quality the right item at the right time, in a rational way. In other words, the decisional subsystem sets and receives orders to schedule the execution of the operations, obeying restrictions such as availability of components or physical devices, minimizing the tardiness of a finished good production and maximizing the use of material or/and resources.

Two of the main functionalities of a decisional system are then addressed: scheduling and disassembling pattern design, as both problems are faced during the design of the control.

2.1. Scheduling

Scheduling consists in allocating tasks to resources and defining dates for the execution of these tasks. Scheduling may be performed for a forward-looking purpose, in order to plan the activities of the resources and to anticipate the achievement of the tasks. Scheduling may also be totally reactive. In this case, the allocation of tasks is performed dynamically and no planning is proposed. Such an approach is proposed in [3] to perform the control of a flexible production system. The lack of mid-term vision, in order to control rationally the manufacturing and to anticipate changes make this approach convenient only for a structurally highly flexible system, like a set of CNC machines. When the sequence of tasks has a high impact on productivity, the planning of manufacturing activities is critical. Thus, we propose a hybrid approach to perform a rational planning and to dynamically alter it in order to respond to disturbances, which may be intentional changes in the environment or random events such as machine breakdowns or product losses.

Solving a scheduling problem aims at minimizing or maximizing one or more criteria, respecting a set of restrictions. A feasible solution is a schedule,

represented by a Gantt chart, which respects the whole set of restrictions. The formal model of a scheduling problem is deeply linked to the characteristics of the system which activity has to be scheduled. Let us consider a task i , modeled by a quadruplet (r_i, t_i, d_i, w_i) where r_i is a release date or earliest start date, t_i is the processing time or operation duration, d_i is the due date or latest finish date and w_i is a weight or priority. Additional restrictions may be related to tasks sequencing or to resource capacity. This model allows formalizing restrictions or to evaluate performance criteria, such as the date achievement of a task for instance.

Most scheduling problems are classified as NP-complete problems [2]. They are highly complex problems, for which the time and computing effort are increasing beyond reasonable limits when the size of the problem increases.

2.2. Disassembling pattern design

The control of the execution of manufacturing activities may address additional optimization problems. Disassembling processes, such as cutting process, require disassembling patterns design (cutting plan) to define the spatial location of the products, in order to be performed. There is a relation between the location of a product in a scheduled disassembling pattern and the date of execution of the disassembling task. Thus, besides scheduling problem, control system should solve optimization problems such as disassembling pattern design [6].

3. Holonic Manufacturing Execution System

A Holonic Manufacturing System (HMS) is a highly decentralized manufacturing system, consisting of autonomous and cooperating agents called holons that respects some flexible control rules forming a holarchy.

The manufacturing execution system (MES) aims at controlling the manufacturing system: what and when to produce, how and when to use the resources, which and when to launch orders. In this section, a holonic manufacturing execution system (HMES) is proposed. The system proposed does not address time and space transformation problems, that is, stock and transport are not considered. Equipments responsible for physical transformation of products are identified as Resource holons, a typology allow to consider:

- Assembling holons,
- Disassembling holons,
- Transforming holons.

Additionally, Supplying holons are in charge of introducing material into the system. The static roles of holons, as well as dynamic mechanisms are presented.

The roles and the models of the three basic holons (Product-Resource-Order) are detailed. These holons interact into two types of holarchies which are each one responsible for an aspect of the manufacturing execution: the availability of the materials and the availability of the resources. Finally, the whole dynamic of this HMES is explained.

3.1. Basic holons definition

At the micro level, individual roles are defined for the basic holons. The class diagram (Fig. 1) shows the relations between the different classes of basic holons as well as their attributes.

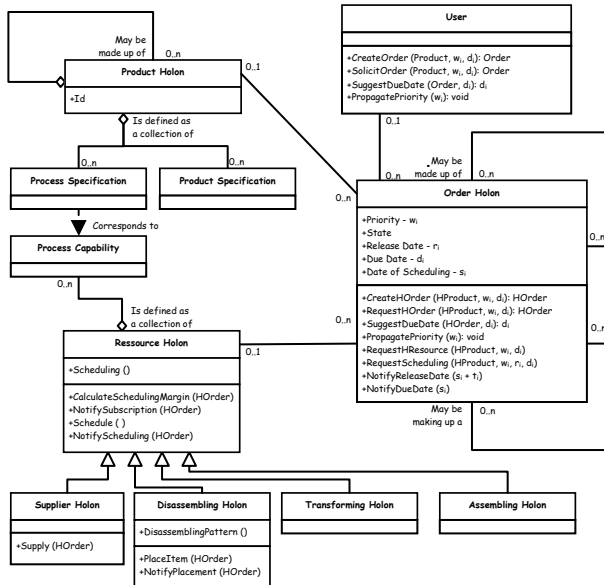


Fig. 1. UML class diagram of basic holons

Each product holon is a product item considered in the production system. Product items define the characteristics and the properties of the products. The bill of material indicates the link of composition between items, this link is instantiated as a reflexive aggregation between Product holons. Product holons are defined as a collection of product specifications which are the physical characteristics of a product (CAD, measurement), and process specifications which describe the process that the component(s) of the product have to undergo. The different models of a product allow handling relevant data to perform the manufacturing control, in function of quality and production processes.

Each physical device considered in the production system with its own logical part is a basic Resource holon. These ones are responsible for lot-sizing and scheduling of the activities of their physical part in accordance with their own model. This model aims at proposing a rational behavior for using material and resource. A typology allows classifying physical devices into four classes, which are specializations of the Resource holon class. Thus, we consider: Supplier

holons, Assembling holons, Disassembling holons and Transforming holons. Resource holons are defined as a collection of process capabilities which correspond to process specifications of product holons. Resource capabilities are production recipes or production methods. The matching between process specifications and process capabilities means that the resource owns the capabilities to manufacture this product. Resource holons are responsible for the scheduling of the tasks of each Order holons. They attempt to schedule these tasks considering release dates, priorities, in function of their capacity and the requirements for each task to be performed (such as tool availability). In addition to the schedule produced, resource holons may produce other results, such as a material cutting plan, in the case of a disassembling process.

Order holons are orders for manufacturing a physical product. Their physical part is a physical product to be manufactured and their logical part contains information about time-restrictions and date of scheduling of the manufacturing operation. Thus, each Order holon is associated to a Product holon which contains the definition of its physical part. The reflexive associations between Order holons model the links of composition between them, as the manufacturing of a product may require components or raw material. These links allow synchronizing the date of the tasks to manufacture a product and its components. To that end, Order holons own a time model of the task required to manufacture its product: each task is defined by a release date (r_i), a due date (d_i), and a date of scheduling (s_i), if the task has been scheduled. Each order holon also has a priority (w_i), which is sorting criteria for each task to be performed. A synchronization of the time restrictions has to be performed so that a due date for an Order holon corresponds to the release date of the Order holon it is a component of.

Order holons subscribe Resource holons which own the specific capabilities to perform the manufacturing of its product. The association between Resource holon and Order holon instantiates this subscription. The Resource holon schedules the Order holons using the attributes of each of them. This time-window approach gives an assessable margin of autonomy for the responsible Resource holon to schedule the tasks. Once the task has been scheduled, the Order holon informs the associated Order holon (upstream and downstream), so as to synchronize the manufacturing of its components and the product of which it is a component of. The synchronization mechanism is described in the next section.

3.2. Holarchies

The interaction between holons allows performing the control of the manufacturing system, which is basically the predictive and reactive scheduling of the

resources (including raw material supply), the predictive and reactive design of the pattern disassembling, and the synchronization of the manufacturing of products and their components. The global behavior of this HMES is the result of the concurrent interaction of holons inside two kinds of holarchie or groups of holons.

Resource holarchies are built up by a single Resource holon and the Order holons which have subscribed to its services. This holarchy aims at performing the scheduling of the activities of a resource and the design of the disassembling pattern if required, the set of information required to control a resource is called a production plan. Local decisional system may be extensive search, complex heuristics or simple priority rules. The use of complex algorithm to take a decision may result in a significant loss of reactivity of the whole system, still priority rules may be used by resource holons while extensive search may be performed by Staff holons. Once a production plan is built and validated, Resource holon notifies the Order holons which are involved in. Plans are designed respecting the time-constraint defined for each Order holon, or delaying the manufacturing task if not possible. This behavior prevents Resource holon from proposing infeasible solutions and thus, the system to face a deadlock. This approach allows embedding any specific local decisional system to control each resource almost independently. The synchronization between Order holons, which is described below, is performed within Order holarchies.

Order holarchies are groups of Order holons which are each one an item product used for the manufacturing of a finished good, or the finished good itself. Each one of these Order holons may be a component of an other Order holon called upper holon, thus, components of an Order holon are called lower holons. The Product holon associated to each Order holon knows the components required to manufacture the product. Nevertheless, associations between Order holons are dynamic, as Order holons are opportunistic and may answer positively to an offer from an Order holon with higher priority and which requires a component. The manufacturing activities require synchronization between Order holons within an Order holarchy. This mechanism aims at guaranteeing that components are available, in order to manufacture a product.

3.3. Dynamic

The whole dynamics of this HMES mainly rely on the activity of the Order holons. From the order creation to the product consumption or delivery to a final client, the Order holon is in charge of soliciting the services of a Resource holon in order to be manufactured, acquiring the components required for its manufacture, and propagating changes of priority

and time limit for its manufacturing. The sequence diagram (Fig. 2) illustrates the working of the whole HMES within a normal mode, and the state diagram (Fig. 3) illustrates the lifecycle of Order holons, which is detailed below.

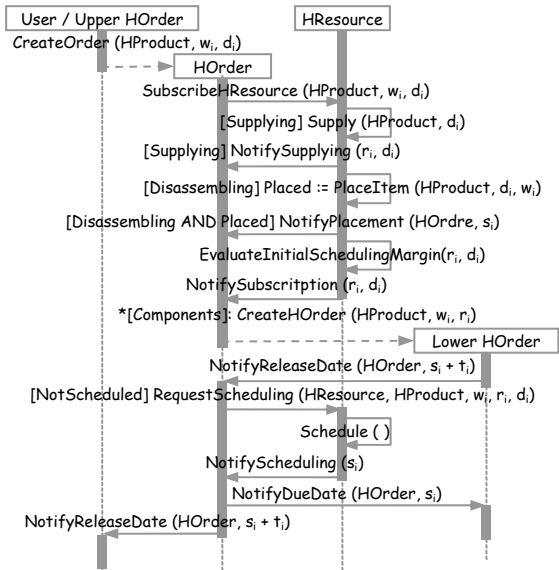


Fig. 2. Sequence diagram of the whole HMES

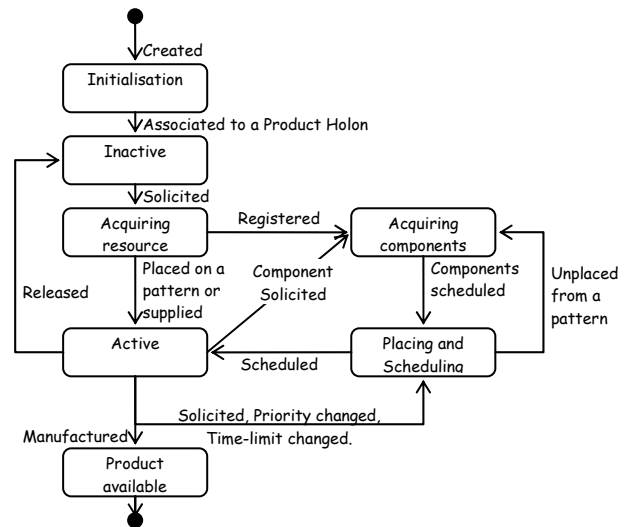


Fig. 3. State diagram of Order holons

The opportunistic behavior of Order holons is also considered as they may respond to a request from a possible upper Order holon with higher priority. In the same way an upper Order holon with a lower priority may lose its components in favor of an other upper Order holon. Order holons are created to respond to a demand concerning a single product, when no existing and available Order holon matches this demand. Such a demand may come from a user (commercial order for a finished product, internal order for stock in process refill), or from an upper Order holon which requires a component in order to be manufactured. Order holons

are initialized with a Product holon which owns the characteristics of the product to be manufactured. Newly created Order holons are automatically solicited; inactive holons are solicited when associated to a user or an upper Order holon. Order holons are then activated when they are informed of a due date, for their item product to be manufactured.

Order holons first query their associated Product holons to retrieve the process specifications required for their manufacturing. Then, they send a request to the Resource holon, which owns the correspondent process capabilities, in order to subscribe its services. To be scheduled, an Order holon must have a release date confirmed. This confirmation happens when its components or lower Order holons are scheduled, since a release date of an upper Order holon is equal to the due date of its lower Order holons. The scheduling mechanism and the respect of the constraints of precedence imply that a task cannot be scheduled if the previous tasks, i.e. the tasks of lower Order holons, are not scheduled. Depending on the type of Resource holon, various cases are described.

If the product is a raw material, no component is required, and the corresponding Resource holon is a Supplier holon. Supplier holons are in charge of raw material stock control; they define a delivery date for raw material. If the required product is in stock, the product may be delivered instantly, whereas the supplier holon may put an order and inform a delivery date to the Order holon. The scheduling is then performed instantly. If the Resource holon is a disassembling holon, it attempts to place the product into an existing disassembling pattern. This pattern must be designed for a lower Order holon which matches the product specification of the component of the subscribing Order holon. If the placement is possible, the subscribing Order holon is associated to this lower Order holon and automatically scheduled.

If the placement fails, and more generally if the Resource holon is a Transforming or an Assembling holon, the scheduling can not be performed instantly, as no confirmed release date is defined for the Order holon. In this case, the Resource holon registers the subscription of the Order holon and evaluates a likely release date for this manufacturing task. This evaluation allows defining a margin of autonomy for the Resource holon to schedule this task, and may be done using a constant quantity of time units, that may depend on the workload of the resource. This mechanism may be considered as planning for manufacturing execution. This likely release date is then informed to the Order holon, as it will be used as a due date for its components.

Once planned, Order holons query other Order holons to complete their holarchy with lower Order holons which match the required components. If no available or opportunistic Order holon responds the

request, Order holons have to create new Order holons for their components to be manufactured. These Order holons are associated to Product holons which own the definition of the required components, and are activated with the likely release date the upper Order holon which created them. This likely release date of the upper Order holon is the due date for the components to be manufactured. Newly created Order holons follow the same lifecycle. This recursive mechanism ends when the components are scheduled and send back to their creator (upper Order holon or user), their date of scheduling. When it happens, the creator of an Order holon has a release date (or a delivery date for the user) confirmed. This creator may then request its Resource holon to schedule its task.

When the release date of a manufacturing task has been confirmed, the corresponding Order holon requests the associated Resource holon to schedule this task and if necessary to place the product into a disassembling pattern. The placing and scheduling are performed by the Resource holon, taking into account the release date, and the priority of each Order holon which has subscribed its services. If the due date is exceeded, the Order holon creator or the user will be advised as the date of scheduling is propagated inside the Order holarchy. During this step, Order holons with lower priority may be unplaced from a disassembling pattern, thus, they should acquire new components before requesting once more placing and scheduling to the Resource holon.

Once scheduled, Order holons propagate their date of scheduling downstream, to confirm the due date of their components, and upstream, for their creator to confirm their release or delivery date. When active, Order holons may be released, due to an Order cancellation, thus, they unsubscribe the Resource holon and go to inactive state, waiting for a new solicitation. This cancellation is propagated downstream to the Order holons components. Order holons components may be solicited by other Order holons with higher priority, their opportunistic behavior drive them to leave their current Order holarchy to enter into a new one. Active Order holons may have to acquire new components when this occurs. Finally, changes of priority and time limit (r_i or d_i), may oblige Order holons to request a rescheduling, and eventually a new placement.

3.4. Scheduling and disassembling pattern design algorithms

As previously described, Resource holons are responsible for the scheduling of activities of a resource. Although the specific behavior of a Resource holon depends on its own model, the scheduling may be performed using a generic rule-based algorithm presented below.

According to section 2.1, each task i is modeled by a release date or earliest start date r_i , a processing time or operation duration t_i (defined by the Resource holon), a due date or latest finish date d_i and a weight or a priority w_i . In order to perform the scheduling, the “earliest by priority order” scheduling rule is used. Tasks to schedule are first sorted by decreasing priority, the Resource holon attempts to schedule them from their release date ; if it is not possible, it attempts to find a slot sliding the task forward (delaying the task).

For disassembling holons, an adaptation of the “earliest by priority” rule is necessary, as various Order holons are gathered together into a single disassembling pattern. In this case, the Resource holon places Order holons by decreasing priority into the disassembling pattern. Patterns are then scheduled using the “earliest by priority” rule. In this case, the priority of a pattern is the highest priority of the Order holon placed in it, and the operation duration is the sum of all the operation durations of the disassembling task of Order holons.

The following algorithm (object-like formalism) implements the “earliest by priority” rule. Let T be the set of tasks i , that have to be scheduled, sorted by decreasing priority (w_i). Once scheduled, set G contains tasks sorted by increasing date of scheduling (s_i).

```

T(1).s=T(1).r
Insert T(1) in G
For i = 2 to n
  j = 1
  /* Looking for a first slot to insert T(i) */
  While {T(i).r > G(j).s} and {j < card(G)}
    j = j + 1
  End while
  /* limit case: T(i) should be scheduled at the end */
  If T(i).r > G(j).s Then j = j + 1
  While T(i) is not scheduled
    If j > card(G) Then /* If the slot to insert T(i) is the last one */
      T(i).s = max(T(i).r, G(card(G)).s+G(card(G)).t)
      Insert T(i) in G after G(card(G))
    Else
      If j = 1 Then /* If the slot to insert T(i) is the first one */
        If T(i).r+T(i).t <= G(j).s Then
          T(i).s = T(i).r
          Insert T(i) in G before G(1)
        Else
          j = j + 1 /* Impossible to insert, let's find the next slot */
        End if
      Else
        /* If the slot is neither the last one nor the first one */
        If max(T(i).r, G(j-1).s+G(j-1).t)+T(i).t <= G(j).s Then
          T(i).s = max(T(i).r, G(j-1).s+G(j-1).t)
          Insert T(i) in G before G(j)
        Else
          j = j + 1 /* Impossible to insert, let's find the next slot */
        End if
      End if
    End while
  End if
End for

```

```

End if
End If
End if
End while
Next i

```

These proposed scheduling rules neither guarantee an optimal solution, nor the respect of time-constraint as latest finish date may be exceeded. Nevertheless, they have very satisfactory performances as they allow retrieving acceptable solutions really quickly. In the case of a due date which is exceeded, Order holon propagates its date of scheduling to its upper holon which will take this date as its new release date to request its own scheduling.

3.5. Processing of disturbances

Disturbances are events which modify the data of the control. These events may be caused by a user (change of priority or due date) or a random event (product loss, or machine breakdown). Some of these events lead to invalidate totally or partially schedules; others do not have any effect on the control, although it may be interesting to reschedule some resources for optimization purpose.

When the schedule is invalidated, the rescheduling is compulsory. Examples of invalidation of a schedule are: subscription of a new Order holon which is impossible to schedule at its release date, increase of the priority of an Order holon which is not scheduled at its release date, delay of the release date of an Order holon that exceeds its scheduling date, delay of the manufacturing of a product which causes an overlap with a task. In these cases, the Resource holon must reschedule. Nevertheless, this rescheduling may affect only Order holons with lower priority than the Order holon responsible for this disturbance. When rescheduling an Order holon, a new date of scheduling is proposed. If this new date is compatible with the latest finish date, it does not cause any disturbance for the upper holon. Whereas if the latest finish date is exceeded, the Order holon advises its upper holon which has to request for a rescheduling. In this case, the disturbance is propagating upstream, until being absorbed by an upper holon or until delaying the delivery to the final client.

A Resource holon may also attempt to reschedule in order to locally optimize the schedule. Examples of events which do not invalidate a schedule are: decrease of the priority of an Order holon, when a release date is put forward, when an Order holon unsubscribes a Resource holon (due to order cancellation), when the manufacturing of a product is performed in advance. These disturbances do not imply to reschedule a resource, since the schedule is still feasible, nevertheless, a rescheduling may be performed for improvement purpose. Alternative schedules may also be proposed by Staff holons. These schedules are

performed using more expensive but more efficient approaches such as complex heuristics or meta-heuristics. These approaches allow taking into account numerous parameters such as a cost function to evaluate a schedule; the result is a better scheduling of the tasks. In a highly perturbed environment, the lifetime of a schedule is shorter, thus the computing time to perform such a schedule may not be sufficient. Then, Resource holons work using their inner rules such as “earliest by priority”. In a relatively less perturbed environment, Resource holon may accept such a schedule as it may propose a better result without introducing disturbances. This schedule is discarded in case of major disturbance.

The events registered from the physical system are not negotiable; they just may or may not lead to modify the schedules and/or the time windows. Various scenarios are conceivable when synchronizing with the physical system. Three main events are considered: achievement of an operation, loss of a product, resource breakdown. Achievement of an operation may happen earlier or later than scheduled, both cases have been considered previously. The loss of a product leads to create a new Order holon which begins a new lifecycle. Resource breakdown may have a major impact on a schedule since it may delay many tasks and compress autonomy margin for the Order.

Disturbances generated or which happen in the system may be absorbed by the autonomy margin of a task. If not, the Order holon for which an inconsistency is detected propagates the disturbance to upper Order holons.

4. Application

The development and implementation of the proposed HMES is motivated by a study performed in a real manufacturing system, which belongs to the American Glass Product (AGP) company.

American Glass Product, produces laminated security glass for automotive application. In the increasingly competitive environment, AGP aims at offering highly customizable products, managing the manufacturing most with design-to-order and make-to-order policy, and in a smaller part with assemble-to-order policy. For this product, quality criteria are demanding as automotive norms are used for quality control. Moreover, the industrialization of the process is recent, and still in progress. The robustness of the process is not guaranteed, and numerous random events occur. Finally, raw materials which go into final product come from high technology and are expensive. A rational utilization of these materials is then necessary. Due to disturbances and mass-customization, numerous normal and degraded flows intermix. The manufacturing policies imply to consider time as a critical parameter. In conclusion, AGP

requires a robust, reactive and flexible system to control costs, delays and quality within its manufacturing system.

The product is defined by its geometry (dimensions, curve), and its composition (glass and plastic layers). The process, presented on the Fig. 4, is split into two main steps: glass pre-process, and glass and plastic assembly. The glass pre-process aims at preparing the glass layers individually: glass cutting, edge grinding and polishing, and silkscreen printing are the operations performed during this step. Glass layers are then gather together as they build up a single product, and are bent in a furnace. This glass structure may be stocked for later use (assemble-to-order). Plastic layers are cut, in function of a customized recipe and the whole is finally assembled. For additional information, an extensive description of the process is available in [1].

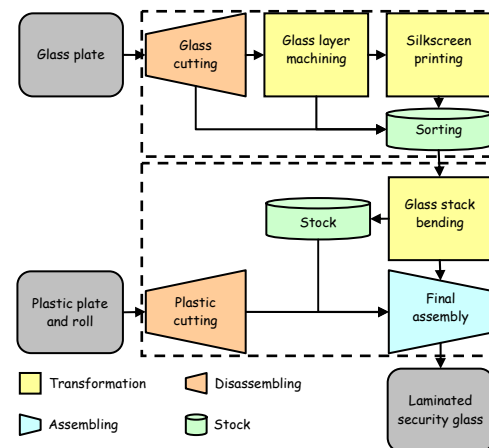


Fig. 4. Synoptic diagram of the process

In this flow-shop manufacturing system, each production cell is autonomously organized, only the sequence of the operations which is materialized as constraints of precedence has to be respected. Supplying, disassembling, transforming, and assembling processes are identified, and decision making model for each corresponding Resource holon is known. These decision making processes are coordinated using the mechanism previously described. The complexity of these models is only limited by the response time. As the holons presented are low-level Resource holons, reactive behavior must prevail. As a result, heuristic rule-based algorithms are proposed to model the decision-making process within each holon. Nevertheless, the use of Staff holons allows integrating any additional model to perform a simultaneous search, including the commercial cutting pattern design software in use today in the firm, which may be encapsulated in a Staff holon.

As an example, a model for a glass cutting holon is presented. In this case, the Resource holon is a Disassembling holon, thus it should solve an allocation and placement problem of the glass layer into a cutting

pattern, and a scheduling problem of the cutting pattern. The allocation problem should first consider the material characteristics (type of glass, color, and thickness) of the glass layer to be cut in order to match with the material characteristic of the stock plates available. The placement problem considers the dimensions of the matching glass plates and the geometric characteristics of each glass layer to be cut, this information is available as a Drawing eXchange Format file (DXF), which is a two-dimensional graphics file format supported by virtually all PC-based CAD products. These sets of information are collections of product specifications controlled by Product holons. Moreover, process specifications for the glass layer holons include rotation restrictions, and minimum trim loss on each side of the rectangle which contains the shape, in order to guarantee a safe cutting process, for both the operator and the glass layer. All this information is made available by way of the Order holon, for the glass cutting holon to control the execution of the manufacturing process. Finally, the cutting glass holon may perform the cutting pattern design and the scheduling of the pattern using the adapted earliest by priority scheduling rule.

The whole HMES is based on the interaction of the micro models of each Resource holon. These models allow controlling the execution of the manufacturing tasks within each Resource holon. As previously described, Order holons subscribe to Resource holons and relay accurate information necessary in order to perform the manufacturing execution control and scheduling. Synchronization between the bounds for each manufacturing task to be executed is realized within the Order holarchies. This communication infrastructure builds a control system from the loose relation of system components.

5. Conclusion

In this article, the use of a holonic approach is proposed to perform the control of a complex and versatile industrial manufacturing system. The use of a holonic structure allows building a reactive and configurable control, while considering global aspects of the manufacturing system. This approach, which is PROSA-based, is compatible with holonic component-based structure.

For this architecture, we propose the use of mechanisms to synchronize the execution of manufacturing orders. These mechanisms guarantee the availability of components and enable to set time limit constraints on task scheduling.

Moreover, a typology of Resource holons is proposed. This typology is based on characteristics of the tasks performed: supplying, disassembling, transforming or assembly. For each type of Resource holons, specific heuristics are used to solve the

manufacturing control problem, taking into account each micro model. For instance, a specific heuristic generates a solution to the scheduling/packing problem in the case of a raw material cutting machine.

These mechanisms and algorithms are implemented within a multi-agents system, which supports the development of the manufacturing control system.

This proposed HMS/MES is valid for discrete-event based manufacturing systems, which consider transformation, assembly and disassembly operations. Possible future development should consider the control of task allocation, when various resources own matching process capabilities. Other developments may reinforce the efficiency of a global scheduling problem solving, propagating scheduling scenarios, within holarchies in order to evaluate them globally. Finally, the synchronization between the manufacturing of the components of a single product may also be addressed in future works.

References

- [1] Blanc P., Demongodin I. and Castagna P., 2006. A Holonic Approach for manufacturing Control: An industrial application. 12th IFAC Symposium on Information Control Problems in Manufacturing, INCOM'06, May 17-19, St Etienne, France.
- [2] Brucker, 2001. Scheduling Algorithms. Springer Verlag, Berlin, 3rd edition, ISBN 3-540-41510-6.
- [3] Bussmann S., Schild K., 2001. An Agent-Based Approach to the Control of Flexible Production Systems. In Proc. of the 8th IEEE Int. Conf. on Emergent Technologies and Factory Automation (ETFA 2001), pp. 481-488 (Vol. 2). Antibes Juan-les-pins, France, 2001
- [4] Deen S.M. (Ed.) (2003). Agent-Based Manufacturing: Advances in the Holonic Approach. Springer ISBN 3-540-44069-0.
- [5] McFarlane Duncan Sarma, Sanjay Chirn, Jin Lung, Wong C.Y., Ashton Kevin, 2003. Auto ID systems and intelligent manufacturing control, Engineering Applications of Artificial Intelligence, 16, pp 365–376.
- [6] Parada V., R.Palma, D.Sales, A.Gómez, 2000. A comparative numerical analysis for the guillotine two-dimensional cutting problem, Annals of Operations Research, 96, pp 245-254.
- [7] Valckenaers P., H. Van Brussel, Hadeli, O. Bochmann, Bart Saint Germain, C. Zamfirescu, 2003. On the design of emergent systems: an investigation of integration and interoperability issues, Engineering Applications of Artificial Intelligence, 16, pp 377–393