HOLONIC MANUFACTURING EXECUTION SYSTEMS FOR CUSTOMISED AND AGILE PRODUCTION – APPLICATION ISSUES

Jean Marcelo Simão - simao@citec.cefetpr.br
Paulo Cézar Stadzisz * - stadzisz@lit.cpdtt.cefetpr.br

Universidade Tecnológica Federal do Paraná (UTFPR)
*Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI).
Laboratórios de Sistemas Inteligentes de Produção (LSIP) e de Inovação Tecnológica (LIT)
Av. Sete de Setembro, 3165 CEP-80230-
901 Curitiba- PR Brazil.

Abstract. *The e-mass-customisation is one of tendencies in production, where which client details products using e-means. This imposes integration and advanced manufacturing-entity collaborations, aiming at adaptability to achieve product heterogeneity and response agility. An integration element of management and shop-floor systems is the Manufacturing Execution System (MES), which may integrate smart-product technology for dealing with customised agile e-manufacturing. In this technology, each product "drives" its manufacture allowing decoupling production and order-dispatching. A smart-product requests and even competes for services of resources, which in turn collaborate based on their features and some flexible logic. However, this collaboration is by itself complex, firstly due to resource heterogeneity. Thus, resources and even smart-products have been "encapsulated" in collaborative entities called Holons (HLs), for homogenisation and integration, thereby contributing for achieving Holonic MES (HMES). HMES comprises also other issues like control of the Holon dynamics. Previous studies proposed a solution, based on "Rules", for control of Holon collaborations presented in the form of a HMES meta-model and applied over the tool ANALYTICE II. This paper proposes a case study about a flexible manufacturing cell, holonified and simulated in ANALYTICE II, which takes into account aspects about agility in customised production and allows proposing solution improvements.*

Keywords: *Mass-customisation, product-rule driven control, HMES, simulation tool.*

1. INTRODUCTION

The mass customisation tendency is a challenge to the Intelligent Manufacturing System (IMS) community (Da Silveira et al., 2001). In fact, customers want to buy products that meet their needs and desires using the easiness of technology (e.g. e-commerce) (Gouyon et al., 2004). This may be partially solved via Manufacturing Execution System (MES) whose purpose is the integration and synergy promotion of management systems and shop floor systems (Morel et al., 2003)(Qiu et al., 2003). However, current MES technologies do not effectively allow customised production (Simão et al., 2006), being necessary to employ concepts like smart-product and Holonic MS (HMS) (McFarlane et al., 2003).

The smart-product concept defines each product coupled with a smart-entity that drives its production, allowing production and order-dispatching independence, as well as physical and informational flows consistency in the MS. A smart-product is able to request services to MS resources (e.g. equipment and work cells) for achieving its production needs. In this context, resources are also improved with some smartness for collaboration, in order to carry out received requests in a more agile way (McFarlane et al., 2003). Nevertheless, the collaboration is itself a problem firstly due to entity heterogeneity.

The entity heterogeneity may be solved by their homogenisation/integration in communicating entities called Holons (HLs). Thus, Smart-Product-HLs collaborate with Resource-HLs composing a HMS, which is by definition agile and able to deal with high-production variety (Van Brussel et al., 1998). However, HMS composition is not trivial because the Holon dynamic may be complex. This motivated studies where the organisation of Holon collaborations is achieved via entities called "Rules", which allow establishing flexible collaboration logic (Simão et Stadzisz, 2002).

Rules decide collaboration moments for potentially requesting Resource-HL services based on states notified by them. In fact, Rules are allocated by Smart-Product-HLs that need operations of a Resource-HL collaboration (Gouyon et al., 2004)(Simão et al., 2006). Thus, the Rule set is a decoupling and organisation mechanism of Smart-Product-HL and Resource-HL collaborations. This solution represents a MES evolution, which has been proposed as a control meta-model for Holonic MES (HMES) and tested in a HMS design and simulation tool called ANALYTICE II (Simão, 2005).

This paper proposes a case study about a Flexible Assembling Cell (FAC), redesigned and simulated in ANALYTICE II, using and improving the proposed control meta-model. In fact, a short-term objective is to make this meta-model an engineering tool for aiding in HMS composition, by using a set of smart-entities and their relationship previously developed and tested. The meta-model would allow reducing system-composition time, being this a reality in ANALYTICE II where experiments allow demonstrating the solution potential.

This paper is structured in the follow manner: section 2 presents the mass customisation issues, HMS/HMES rationales and the HMS design/simulation tool ANALYTICE II. After that, section 3 presents a solution for process and product driven control for HMES/HMS. Section 4 presents a case study about the solution application over the FAC. Finally, section 5 presents conclusion and foresees future works.

2. HMS/HMES FOR MASS CUSTOMISATION

For dealing with varying production, manufacturing organisations must exploit their own flexibilities. Thus, researches propose auto-organised MS entities, improving manufacturing processes (Deen, 2003)(Morel et Grabot, 2003). Classically, production is planned in lots, via Enterprise Requirement Planning (ERP) systems, based on previous client demands, where entities (e.g. controllers and machines) are prepared to produce few product types in a period. This policy is not interesting to mass customisation issues once response time can be too long.

A solution is each order, about a product, be a smart-entity that knows capacities and states of advanced resources (i.e. flexible and configurable ones) and suitably launches its own production. The resources are also enhanced with some expertise to allow smart-orders knowing their states/capabilities and requesting their services (McFarlane et al., 2003). This may be accomplished by attaching an agent to each resource, via computational-electronic means as represented in Fig. 1, adapted from (Hartley, 1984). The physical resource and its agent are seen together as a smart-resource (Simao et al., 2006).

In this smart MS, smart-orders negotiate with smart-resources and launch their own production. This order-driven approach eases collaborations once production actors and their negotiations are homogenised at software level. But, questions remain such as possible incoherencies, e.g. a smart-order may believe that its concerned product is in a given place but it is not. A solution is each smart-order integrated with its product, being a smart-product. A way is to identify the product with some frequency and update the correspondent agent, which can be made via RFID (Radio Frequency Identification) technology (McFarlane et al., 2003).

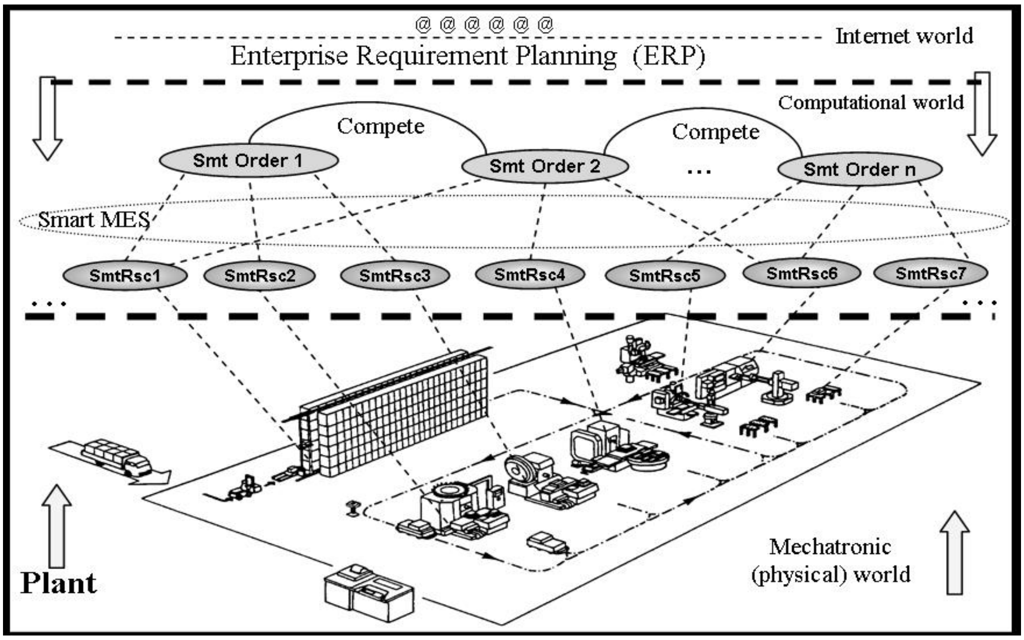


Figure 1. Integrated environment with Smart-Resources and Smart-Orders

The smart entity concept is related to the Holon concept. A Holon is an autonomous and collaborative MS building block for transporting, storing, and/or validating information and physical objects. It consists of an information processing part and often a physical part (Van Brussel et al., 1998). In fact, the most focused approach in the IMS researches is the holonic paradigm, originated from a philosophical theory on the creation/evolution of world adaptive systems (e.g. social systems). The main idea is to achieve good properties of natural systems (e.g. adaptability and flexibility) in MS (Morel et Grabot, 2003)(Valckenaers, 2001).

A Holonic MS (HMS) is based on Holon collaborations, namely on Smart-Product-HLs and Resource-HLs. However, they negotiating in heterarchical way are not enough once problems may appear, e.g. states unpredictability, deadlock or states explosion) (McFarlane et al., 2003. Thus, a MES-like is necessary to control their collaborations ensuring operability/adaptability by avoiding strong hierarchism, i.e. a trade off between heterarchy and hierarchy forming a holarchy. In fact, this MES-like is an industrial and IMS-community concern (Qiu et al., 2003)(Morel et al., 2003)(Van Brussel et al., 1998).

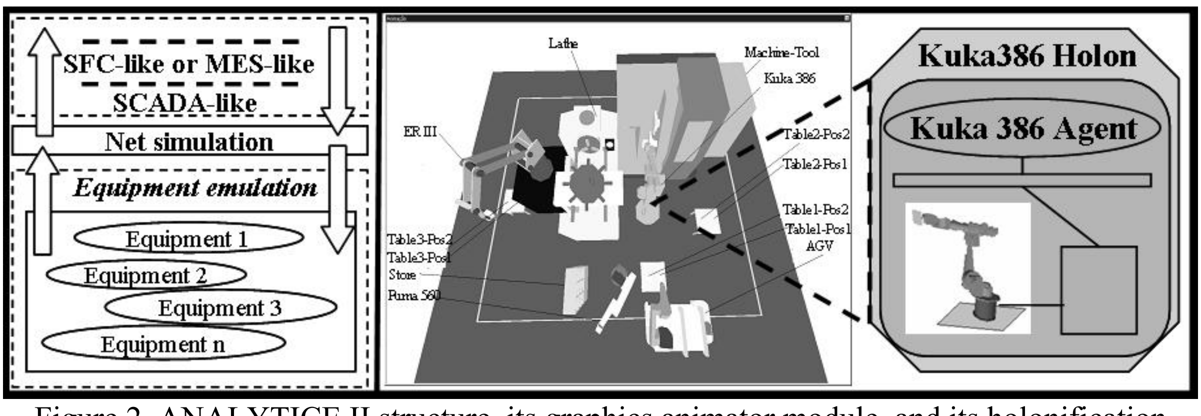


Figure 2. ANALYTICE II structure, its graphics animator module, and its holonification

The next sections present efforts in a Holonic MES (HMES) solution. This solution is validated over the singular HMS design/simulation tool ANALYTICE II. It was developed at LSIP/UTFPR for CIM (Computer Integrated Manufacturing) issues, but its primitives allowed its holonification (Simão, 2005). Like in real MS, it separates the execution of resources and Shop-Floor-Control (SFC, a MES synonymous) via a ‘virtual’ network (Fig. 2). Thus, an agent receives signals from and requests services for each resource, via the network, forming realistic Resource-HLs. In fact, in ANALYTICE II or real MS, Resource-HLs substitute a SFC/MES part usually called of SCADA (Supervisory Control and Data Acquisition).

3. HOLONIC CONTROL SOLUTION

The holonic control or HMES solution starts by evolving Resource-HLs. In this solution, Resource-HLs express resource states by Attribute subagents and receive services demands via Method subagents, homogenising their work manner and then facilitating control activities. The first MESs over these Resource-HLs were process-driven controls that allowed generated an architecture (Simão et Stadzisz, 2002). The solution was inspired by Rule Base System (RBS) being each instance a type of Expert System (ES) to carry out the control of Resource-HL collaborations (Simão et al., 2003).

In fact, each control instance is an ES-like whose fact base is related to states of Resource-HL Attributes, the decision and coordination is carried out by Rules and the final conclusion is instigations of Resource-HL Methods. The Rules and their associated entities (e.g. Premises and Methods) are also Holons (Gouyon et al., 2004)(Simão et al., 2006). Figure 3 presents an example of Rule knowledge and a Rule class diagram in UML (Unified Modeling Language). Two rule-driven control instances, over ANALYTICE II, are described below (Simão, 2005).

The production of parts X and Y was simulated in the Fig. 2 system. The Production Plan (PP) for X was {<Store Pos1…6><Table1 Pos2><Machine><Table2 Pos1…2>} and for Y was {<Store Pos7…9><Table1 Pos2><Table3 Pos1><Lathe><Table3 Pos2>}. A Rule set was created allowing Resource-HL collaborations to produce. Also, a third product type was introduced and the second type was enabled to use a second added lathe. This led to change PP and Rule sets, allowing observe that: system adaptability for agility is feasible by changes of Rule-knowledge being, however, needed to validate Rules, e.g. by simulation.

It was still simulated in that system the production of a real part from AIPL (*Atelier Inter-établissements de Produtique – Lorraine*), a training place related to *Centre de Recherche en Automatique de Nancy* (CRAN), described in (Gouyon et al., 2004). The PP to this part is {<Table3 Pos1><Lathe><Table3 Pos2><Table1 Pos1…2><Machine><Table2 Pos1…2>}. A Rule set was also used and statistical results were taken. Briefly, the productivity was 83.68%, meaning only that Resource-HL loading and unloading time must be optimised. In the control viewpoint, Rules made their function, i.e. control Resource-HL collaborations.

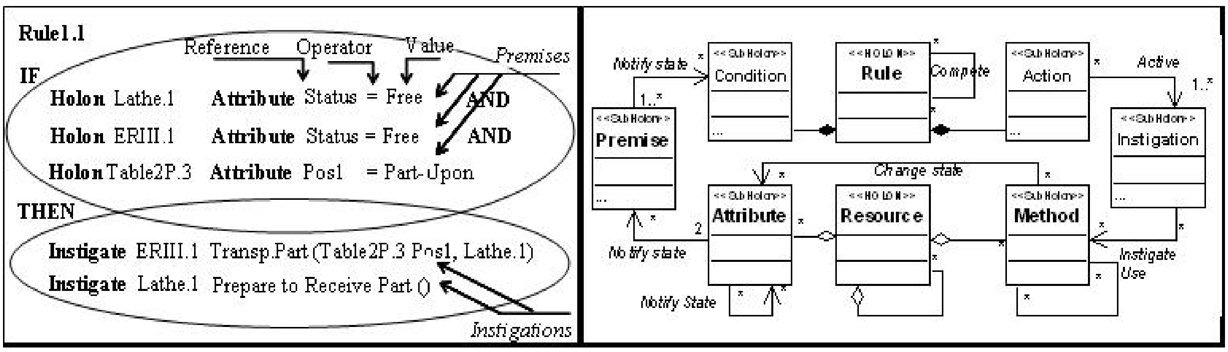


Figure 3. Rule and its associated entities

The agents-notification (Fig. 4, left-side) is more than an elegant control solution. It represents an inference engine that eliminates search, brings a quicker reactivity, and allows means to identify and resolve rule conflicts. The solution elaboration also takes into account good practices of system engineering, such as functional independence between entities and trade-off between generality and applicability. Furthermore, specific control, computational, and systemic concerns were considered in its construction and evolution (Simão, 2005).

The control concerns are: (a) determinism and reactivity trade-off; (b) conflict identification and resolutions via notifications, and (c) openness to formalism via Petri net compatibility (Simão et al., 2003). The computational concerns are: (a) performance via redundancy avoidance and (b) openness to distribution via functional independence (Simão et Stadzisz, 2002). The systemic concerns are: (a) instance adaptability, e.g. MS-flexibility use by suitable Rules; (b) human integration, e.g. Rules are intuitive elements; and (c) openness to systemic integration, e.g. solution improvement for be product-driven.

In fact, in the case of product-driven, the improvement is "simple": Smart-Product-HLs allocate Rules according to its needs (Fig. 4). Thus, a Rule execution depends also of their allocation. The solution was tested in case studies, like the MS presented above. The main advantage, in short, is better adaptability/agility for customisation profiting Rule gains that, in this context, is organisation/optimisation of Holon collaborations (Simão et al., 2006). This paper details how to apply this approach over a Flexible Assembly Cell (FAC) and investigates the potential benefits.

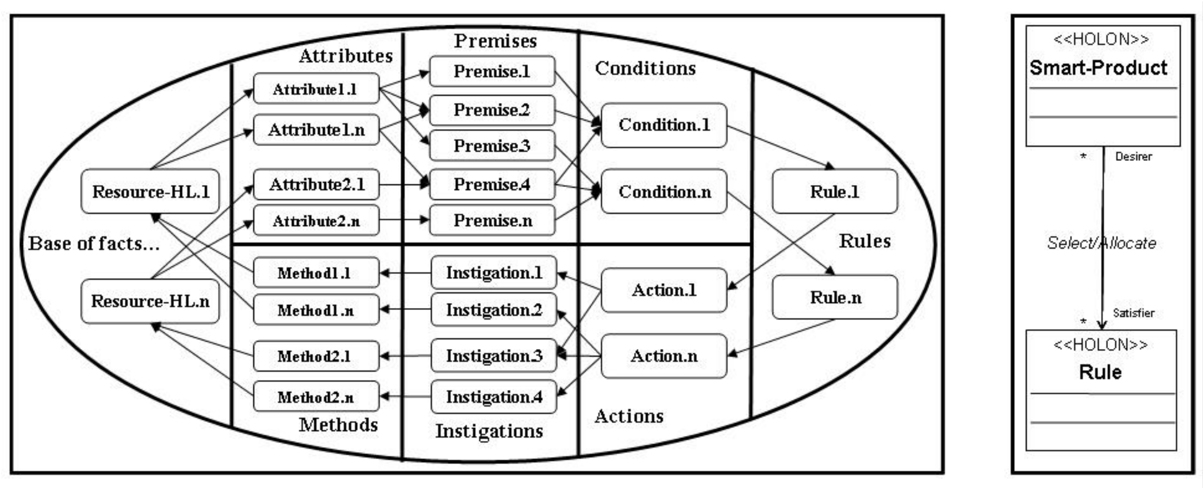


Figure 4. (a) Notification mechanism; and (b) Solution upgraded for product-driven.

4. CASE STUDY - HOLONIC FLEXIBLE ASSEMBLY CELL (HFAC)

The FAC (Fig. 5), from AIPL, provides a real MS for engineering students. It can assemble six pedagogical product types from six part types buffered in certain workstations (WSs). There are a WS for loading pallets that circulates on a conveyor, four WSs for assembling products on pallets stopped in face, and one WS for unloading products. The conveyor has segments for pallet buffering and stopping/identification positions (Fig. 6) for deciding if a pallet have to visit a given WS or not. Each pallet is able to carry four products and has a digital memory for product information (Gouyon et al., 2004).

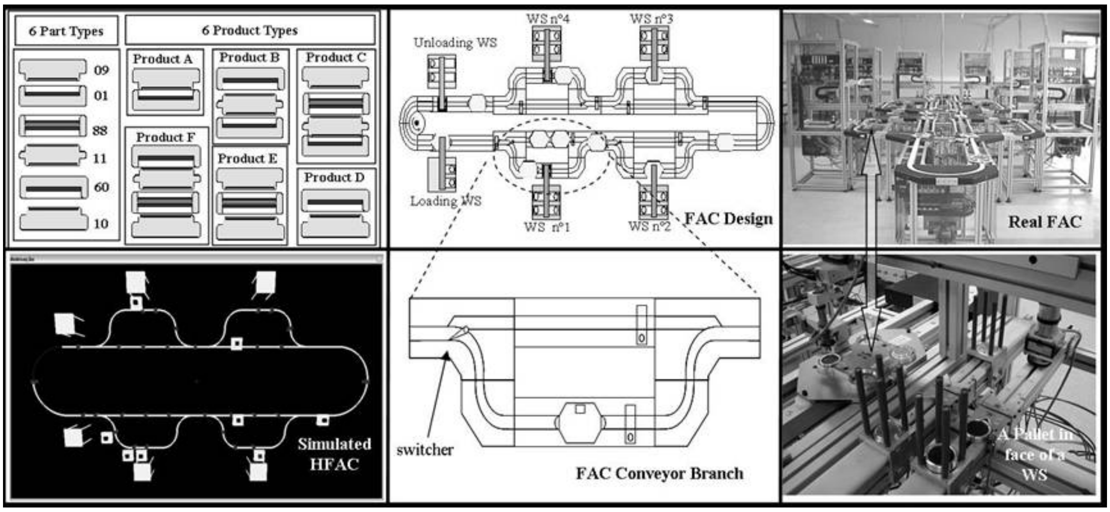


Figure 5 – The FAC (Flexible Assembling Cell) and products types made therein.

The FAC was redesigned as a Holonic FAC (HFAC) with Smart-Product-HLs and Resource-HLs. Base parts are passive Smart-Product-HLs whereas assembled products are active ones for production driving. The Resource-HLs are Conveyor-HL, WS-HLs, and Pallet-HLs named as: (a) Conveyor.1; (b) WorkStation.0 … and WorkStation.5; and (c) SmPallet.1 … and SmPallet.10. These Resource-HLs were implemented and simulated in ANALYTICE II, by means of emulated-resources and respective virtual-resources.

Each emulated WS is as a *black-box* that receives command and gives feedback after specific time, being graphically a block. Each WS-HL has a main Attribute (Status) and Commands for assemblages. In turn, the emulated-conveyor graphically and logically transports pallets with products. The Conveyor-HL has Attributes for buffering regions and positions and Commands for requesting pallet releases and switcher moving, about positions.

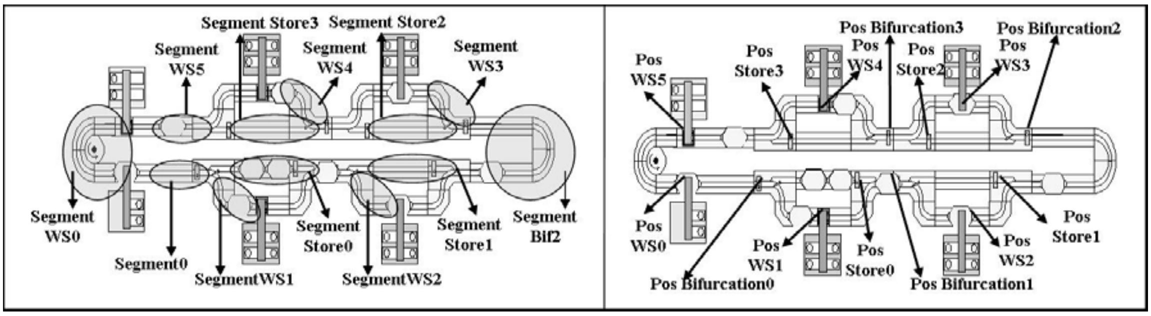
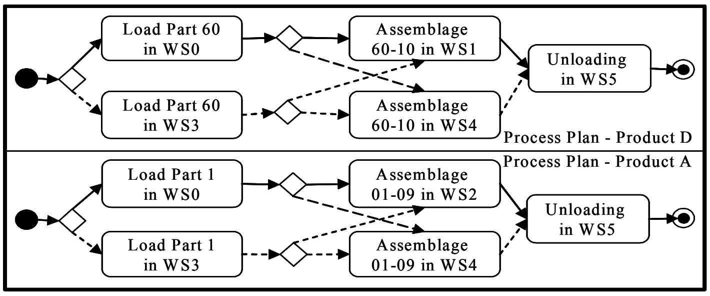


Figure 6 - Segments and positions in the conveyor.

In the HFAC, Process Plans (PPs) define the WS-HLs that must be visited by Smart-Product-HL types. The PP creation starts by foreseeing base-part types to WSs, for example: WS0 – part types 01 and 60; WS1– 10, 11, and 88; WS2 – 88, 09, and 11; WS3 – 88, 60 and 01; and WS4 – 11, 10, and 09. After that, PPs with alternative paths are defined for all product types, being PP-A and PP-D presented in Fig. 7. In fact, PP-A and PP-D were formally defined by a method based on theory of automatic synthesis (Gouyon et al., 2004).

These PPs specify processing resources, but do not buffering or transport ones. However, the identification of all (part of) concerned Resource-HLs ease Rule creations. Thus, PP-D and PP-A main-branches were detailed, being PP-A presented as example: **(1)** <u>WS0 Load Part.60.</u> **(2)** Transfer from WS0 to Store0. **(3)** Wait in Store0. **(4)** Transfer - Store0 to WS2. **(5)** <u>WS2 Assemblage 01,09.</u> **(6)** Transfer - WS2 to Store2. **(7)** Wait in Store2. **(8)** Transfer - Store2 to Store3. **(9)** Wait in Store3; **(10)** Transfer - Store3 to WS5. **(11)** <u>WS5 Unloading</u>.



Rule As0 If WorkStation.0 Status = Pallet-OK Conveyor.1 PositionWS0 = Pallet-Upon Conveyor.1 Segment0 = Have-Space Then Conveyor.1 Liberate_Pallet(PositionWS0)
Rule As5 If WorkStation.5 Status = Pallet-OK Conveyor.1 Segment0 = Have-Space Then Conveyor.1 Liberate_Pallet(PositionWS5)

Figure 7 – (a) Process Plan for D and A Products – (b) First Rule set about the HFAC.

A Rule set was created based on PPs, for allowing control, using the same structure from other experiments. The difference was only Rule knowledge (Attributes linked to Premises and Methods linked to Instigations) now concerned to HFAC. For example, the first Rule in Fig. 7 deals with the moment to free a pallet from PosWS0 whereas the second with the moment to free a pallet from PosWS5. A Smart-Product-HL can use Rule As0 for reaching an operation and a Pallet-HL can use Rule As5 for reaching PosWS0. The difference is the explicit Smart-Product-HL allocation of As0 and the Pallet-HL passivity about As5.

The Rules in Fig. 8, in turn, are divided in four groups (As1 … and As4). Each one firstly allows a Smart-Product-HL chooses between a Rule type As*X*.A0 or As*X*.B0. The first allows it to reach a WS and the second a Store. If an As*X*.A0 is chosen then the next is an As*X*.A1 for leaving WS. Alike, after an As*X*.B0, there is an As*X*.B1 for leaving the Store. The suitable Rule allocation by Smart-Product-HL depends of their PPs. Anyway, these presented Rules allow contemplating PPs of all types of Smart-Product-HLs foresee to the HFAC.

The HFAC simulated Smart-Product-HLs of type A and D, considering PP first-branches. Each type was graphically represented by a cube with a specific color for observing their behavior and then Rule allocations. The base parts and assemblages were not graphically simulated whereas loading/unloading operations were. Functionally, all were simulated being operations started via Rules allocated by Smart-Product-HLs. An interesting fact was Smart-Product-HL creation without a physical part. Only after the loading it acquires a part that is not graphically changed but virtually updated, after assemblages, in Smart-Product-HL.

Rule As1.A0 If Conveyor.1 PosBifurcation0 = Pallet-Upon Conveyor.1 SegmentStore0 = Have-Space Then Conveyor.1 Move_Switcher0(SegmentStore0) Conveyor.1 Liberate_Pallet(Bifurcation0)	Rule As1.B0 If WorkStation.1 Status = Free Conveyor.1 PosBifurcation0 = Pallet-Upon Conveyor.1 SegmentWS1 = Have-Space Then Conveyor.1 Move_Switcher0(SegmentWS1) WorkStation.1 Prepare()	Rule As2.A0 If Conveyor.1 PosBifurcation1 = Pallet-Upon Conveyor.1 SegmentStore1 = Have-Space Then Conveyor.1 Move_Switcher1(SegmentStore1) Conveyor.1 Liberate_Pallet(Bifurcation1)	Rule As2.B0 If WorkStation.2 Status = Free Conveyor.1 PosBifurcation1 = Pallet-Upon Conveyor.1 SegmentWS2 = Have-Space Then Conveyor.1 Move_Switcher1(SegmentWS2) WorkStation.2 Prepare()
Rule As1.A1 If Conveyor.1 PosBifurcation1 = Free Conveyor.1 PosStore0 = Pallet-Upon Then Conveyor.1 Liberate_Pallet(PosStore0)	Rule As1.B1 If WorkStation.1 Status = Pallet-OK Conveyor.1 PosBifurcation1 = Free Then Conveyor.1 Liberate_Pallet(PositionWS1)	Rule As2.A1 If Conveyor.1 PosBifurcation2 = Free Conveyor.1 PosStore1 = Pallet-Upon Then Conveyor.1 Liberate_Pallet(PosStore1)	Rule As2.B1 If WorkStation.2 Status = Pallet-OK Conveyor.1 PosBifurcation2 = Free Then Conveyor.1 Liberate_Pallet(PositionWS2)
Rule As3.A0 If Conveyor.1 PosBifurcation2 = Pallet-Upon Conveyor.1 SegmentStore2 = Have-Space Then Conveyor.1 Move_Switcher2(SegmentStore2) Conveyor.1 Liberate_Pallet(Bifurcation2)	Rule As3.B0 If WorkStation.3 Status = Free Conveyor.1 PosBifurcation2 = Pallet-Upon Conveyor.1 SegmentWS3 = Have-Space Then Conveyor.1 Move_Switcher2(SegmentWS3) WorkStation.3 Prepare()	Rule As4.A0 If Conveyor.1 PosBifurcation3 = Pallet-Upon Conveyor.1 SegmentStore3 = Have-Space Then Conveyor.1 Move_Switcher3(SegmentStore3) Conveyor.1 Liberate_Pallet(Bifurcation3)	Rule As4..B0 If WorkStation.4 Status = Free Conveyor.1 PosBifurcation3 = Pallet-Upon Conveyor.1 SegmentWS4 = Have-Space Then Conveyor.1 Move_Switcher3(SegmentWS4) WorkStation.4 Prepare()
Rule As3.A1 If Conveyor.1 PosBifurcation3 = Free Conveyor.1 PosStore2 = Pallet-Upon Then Conveyor.1 Liberate_Pallet(PosStore2)	Rule As3.B1 If WorkStation.3 Status = Pallet-OK Conveyor.1 PosBifurcation3 = Free Then Conveyor.1 Liberate_Pallet(PositionWS3)	Rule As4.A1 If Conveyor.1 SegmentWS4 = Free Conveyor.1 PosStore3 = Pallet-Upon Then Conveyor.1 Liberate_Pallet(PosStore3)	Rule As4.B1 If WorkStation.4 Status = Pallet-OK Conveyor.1 SegmentWS4 = Have-Space Then Conveyor.1 Liberate_Pallet(PositionWS4)

Figure 8 – Second Rule set about the HMES of the HFAC.

In the HFAC, after a Smart-Product-HL has physical and virtual parts, they are regularly synchronized for avoiding inconsistence. When the pallet arrives in a conveyor position, the conveyor reads its identification and sends the information to the virtual-net, which serves to update the virtual-pallet and associated virtual-product set. In this case, eight Pallet-HLs were used for Smart-Product-HLs transporting, being one by Pallet-HL. This simplifies synchronizations and 'avoids' merge problems about Smart-Product-HL desires.

Smart-Product-HLs may use main desired Rules or alternative ones for exploiting MS flexibilities. For testing the use of alternative Rules, the SegmentWS1 was loaded forcing Smart-Pallet-HLs to get an alternative one. For example, a Smart-Product-HL of type D at the PosBifurcation0 tries to select/allocate As1.B0 for reaching WS1, but it is false (segment full), then it searches a second option, As1.A0. Once allocated, this Rule allows it going ahead, suitably selecting/allocating subsequent Rules (As1.A1, As2.A0, As2.A1, As3.A0, and As4.B0) for achieving the alternative production resource, WS4.

Actually, in HFAC, resources coordination may happen via an interlocked physical system. However, resource real-time information via Resource-HLs notifying Rules allows Smart-Product-HLs to infer and choose better production means. An example is the more equilibrated WS use, where each Rule evaluates "if a segment of a WS is semi-loaded and if another of an alternative WS is non-loaded" that allows balance by Smart-Product-HLs allocating alternative Rules. Other example is two Smart-Product-HLs competing for a Rule (Fig. 9), being winner the priority one that forces the other to search for an alternative Rule.

The control solution applied for HFAC does not change structural MS bases. It augments MS information, integration, and control level aiming at better MS use. Rule and Smart-Product-HL knowledge is based on parameters created for achieving a performance goal. The holarchy can set Holon knowledge aiming to optimize resource use, improve benefits with more lucrative products, reduce times for agility, deal with customization, and so forth. This case study, even with potential quantitative features (e.g. 73% in average use of WSs), is mainly qualitative for observing solution potential for different performances goals.

In FAC-like cases, the solution advantage is actually informational support for production decisions. This advantage may be still improved with additional mechanism possible due to the approach based on notifications (Simão, 2005). A suitable example is Rule approbation prevision. In Fig. 9 (right side), a case is shown where a Smart-Product-HL allocates an alternative Rule once the main has a *false state* that, however, achieves a *true state* a moment later. If an approbation prevision is enabled, it could wait a little time for a better Rule use.

For carry out this mechanism, each Attribute must foresee a new state and notify related Premises, e.g. WorkStation.1 Status knows that a *free* state will be achieved in 3 seconds and notifies it with 97% of certainty. In turn, each notified Premise makes its logic calculus about a new state and those 'changed' notify related Rules. Each notified Rule calculates an approbation prevision when all related Premises, in false state, notified a future true state. The time/probability prevision may be useful then for decisions of a Holon set. The point is that this approach has a set of properties for different performance goals, such as customization.

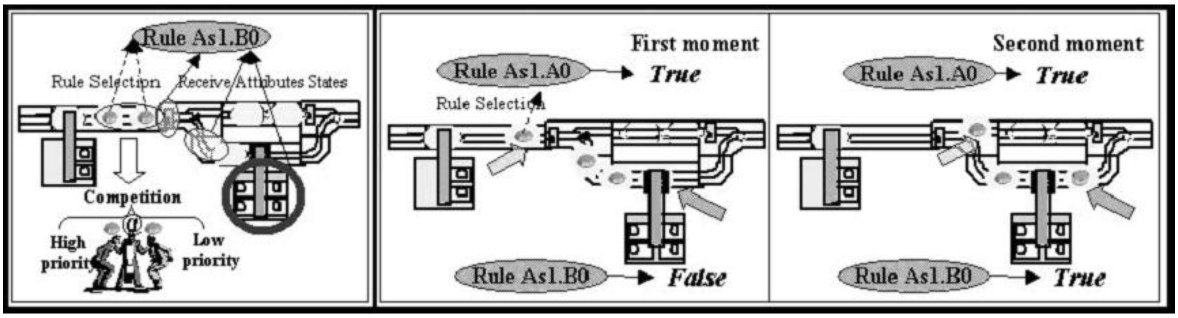


Figure 9 – Flow issues.

5. CONCLUSIONS

This paper presents a control approach for Holonic MES of HMS. This solution starts with the resource holonification, based on computational homogenisation, which generates Resource-HLs with expertise for expressing their capabilities/states and receiving service requests. This expertise allows, in fact, Resource-HLs carrying out control functions, namely monitoring and command. Subsequently, Rules are presented as organisers of Resource-HL collaborations, allowing process-driven HMES. The Rules are also Holons that work based on a previously proposed inference process: a notification-oriented method.

The paper follows in the Smart-Product-HL presentation, regarding mass customisation issues. A Smart-Product-HL indirectly reserves Resource-HLs allocating suitable Rules. Each Rule correctly coordinates Resource-HL services to carry out a Smart-Product-HL desire. Therefore, a solution to product-driven HMES is given by these Holon collaborations regulated by Rules. In the product-driven context, Rules are a decoupling mechanism between Smart-Product-HL and Resource-HL collaborations, allowing organise and optimise them.

This presented control architecture has its first practical contribution in the ANALYTICE II holonification, easing simulated-HMS composition. In fact, holonic control experiments have been developed therein, demonstrating the solution "meta" feature (generality). As result, it is considered a meta-model to HMES, firstly in this simulation environment. The solution can be understood as an actual one for real HMSs once similar Resource-HLs and Smart-Product-HLs have been developed in IMS community, namely by McFarlane (2003).

On a mass-customisation point of view, this paper presents HMES tools to support and examine product-driven benefits. The results agree with those from IMS literature, namely the capacity to produce possible product-types without prevision. An advantage of the presented approach is organisation and information enable by Rules and their notification net. These features were particularly observed via the proposed HFAC case study, which is an additional control meta-model application.

The HFAC case study has confirmed the approach properties and allowed to investigate new details, such as the association between logical and physical parts in the case of assemblages. Moreover, the case study has highlighted that flexibility exploitation for agility/customization depends on enabled information and presented mechanisms for achieving better plant-information availability and utilization. Namely, the approbation prevision mechanism was described as an improvement of the notification-mechanism.

The foreseen works includes a deep development and evaluation of case studies about simulated and real applications, including control-solution evolutions. An example of case study, from Simão (2005), is about AIPL simulation where HFAC is an Assembling-Cell-HL related to other Cell-HLs in a just-in-time production context. This experiment studies the control meta-model application as an HMES in a plant, beyond manufacturing cells. In addition, other works are being realised about larger solution use, highlighting application and advantages of the notification mechanism for inference and discrete systems in general.

6. ACKNOWLEDGEMENTS

J. M. Simão would like to thank CAPES/Brazil (*Coordenação de Aperfeiçoamento de Pessoal Nível Superior*) for national/international PhD. scholarships between May 2001 and April 2004. The authors would like to thank structure and resources enabled by LSIP, CPGEI, CEFET-PR/UTFPR, UHP, CRAN, and AIPL. The authors would like to thank Prof. G. Morel for collaboration at CRAN/UHP. Finally, the authors would like to thank the ANALYTICE II authors (Koscianski et al., 1999), especially L. F. F. Rosinha (*in memoriam*) that additionally helps to develop the emulated equipment for HFAC.

7. REFERENCES

Da Silveira, G., Borenstein, D., & Fogliatto, F.S., 2001. Mass customization: literature review and research directions. *International Journal of Production Economics*, vol. 72, pp 1-13,

Deen, S.M, 2003. Agent-Based Manufacturing: Advances in the Holonic Approach. Springer. ISBN 3-540-44069-0.

Gouyon, D., Simão, J. M., Khaled, A, & Morel, G., 2004. Product-driven issues for B2M-control systems integration. In: *INCOM 2004 –11th IFAC Symposium on Information Control Problems in Manufacturing*, Salvador – BH, Brazil.

Hartley, J, 1984. FMS at work. IFS (Publications) Ltd., UK. ISBN 0-903608-62-6.

Koscianski, A, Rosinha, L. F. F., Stadzisz, P. C., & Künzle, L. A, 1999. FMS design and analysis: developing a simulation environment. In: *Proceedings of the 15th Intern. Conf. on Cad/Cam, Robotics and Factories of the Future, Águas de Lindóia*, Brazil, vol. 2.

McFarlane, D., Sarma, S., Chrin, J. L., & Ashton, K., 2003. Auto ID systems and intelligent manufacturing control. *Intelligent Manufacturing: Special issue of Engineering Applications of Artificial Intelligence*, vol. 16, n. 4, pp. 365-376.

Morel, G. & Grabot, B., 2003, (Eds.). *Special issue of Engineering Applications of Artificial Intelligence*. Intelligent Manufacturing, vol. 16, n.4.

Morel, G., Panetto, H., Zaremba, M. B., & Mayer, F, 2003. Manufacturing Enterprise Control and Management System Engineering: Rationales and Open Issues. *Annual Reviews in Control*.

Qiu, R., Wysk, R., & Xu, Q, 2003. Extended structured adaptive supervisory control model of shop floor controls for an e-manufacturing system. International Journal of Production Research, vol. 41, n.8, pp. 1605-1620, 2003, ISSN 0020-7543.

Simão, J. M., 2005. *A Contribution to the Development of a HMS simulation tool and Proposition of a Meta-Model for Holonic Control*, Ph. D. Thesis, Federal University in Technology of Paraná (CPGEI/UTFPR) & Henry Poincaré University (CRAN/UHP).

Simão, J. M. & Stadzisz P. C, 2002. An Agent-Oriented Inference Engine applied for Supervisory Control of Automated Manufacturing Systems. *Frontiers in Artificial Intelligence and Applications: Advances in Logic, Artificial Intelligence and Robotics LAPTEC 2002*, vol. 85, pp. 234-241, IOPRESS, Amsterdam, ISBN I 58603 292 5.

Simão, J. M., Stadzisz, P. C., & Morel, G, 2006. Manufacturing execution system for customised production. *Journal of Material Processing Technology*, vol. 179, issues 1-3, Elsevier (from 3rd COBEF).

Simão, J. M., Stadzisz P. C., & Künzle L. A., 2003. Rule and Agent-Oriented Architecture to Discrete Control Applied as Petri Net Players. In: *Frontiers in Artificial Intelligence and Applications: Advances in Intelligent Systems and Robotics LAPTEC 2003*. vol. 101, pp. 121-129, IOS PRESS, Amsterdam - The Netherlands, ISBN I 58603 386 7.

Valckenaers, P, 2001. *Special issue: Holonic Manufacturing Systems*. Computers in Industry, vol. 46, issue 3, pp. 233-331.

Van Brussel, H., Wyns, J., Valckernaers, P., Bongaerts, L., & Peeters, P., 1998. Reference architecture for holonic manufacturing systems PROSA. *Computers in Industry*, vol. 37, pp. 255-274.