



HAL
open science

MODULARITY OF RMS: WHAT ABOUT THE AGGREGATION?

Erica Capawa Fotsoh, Pierre Castagna, Nasser Mebarki, Pascal Berruet

► **To cite this version:**

Erica Capawa Fotsoh, Pierre Castagna, Nasser Mebarki, Pascal Berruet. MODULARITY OF RMS: WHAT ABOUT THE AGGREGATION?. 13ème CONFERENCE INTERNATIONALE DE MODELISATION, OPTIMISATION ET SIMULATION (MOSIM2020, Nov 2020, AGADIR (virtuel), Morocco. hal-03177750

HAL Id: hal-03177750

<https://hal.science/hal-03177750>

Submitted on 23 Mar 2021

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.

MODULARITY OF RMS: WHAT ABOUT THE AGGREGATION?

Erica CAPAWA FOTSOH^{1,2}, Pierre CASTAGNA²

¹ IRT Jules Verne (French Institute in Research and Technology in Advanced Manufacturing), 44340 Bouguenais, France (erica.fotsoh@irt-jules-verne.fr, erica.fotsoh@ls2n.fr).

² Nantes University, IUT of Nantes, LS2N, BP539 44475 Carquefou Cedex, France
pierre.castagna@univ-nantes.fr

Nasser MEBARKI², Pascal BERRUET³

³ Lab-STICC Research Center, University Bretagne-Sud, BP 92116 56321 Lorient, France
nasser.mebarki@univ-nantes.fr
pascal.berruet@univ-ubs.fr

ABSTRACT: *Modularity of Reconfigurable Manufacturing System (RMS) reduces the system complexity, both in design and evaluation of the system. In this paper we proposed a modular view of the system that considers a module regarding four aspects: physical, control logic, KPI and simulation model. The last two one allows an accurate diagnosis of the system in case of disturbances. Each module is evaluated and studied. Yet, the reconfiguration decision is made regarding the system level, thus, modules have to be aggregated. The lack of aggregation methods for simulation model and control logic, and the lack of aggregation functions for KPIs constitute the issue addressed by this paper. The aim of our study is to investigate the aggregation of these aspects of the modules in order to have accurate data at the system level for reconfiguration decision making. A case study in conducted to show how aggregation could be made. Far from proposing rigid models of aggregation, the aim of the paper is to provide opportunities of thinking about the notion of aggregation of simulation models, KPI and control logic within manufacturing context, especially within modular reconfigurable manufacturing system.*

KEYWORDS: *Modularity, Aggregation methods-functions, simulation model, KPI, RMS, reconfiguration decision-making*

1 INTRODUCTION

Reconfigurable manufacturing systems (RMSs) are designed to produce “what is needed just when is needed”. (Mehrabi et al., 2000). This is possible because the configuration chosen for the system at a given time matches with the needs that arise. A deep diagnosis of the system is often needed to clearly identify the reconfiguration needs and to determine an appropriate reconfiguration strategy (Capawa Fotsoh et al., 2020). This involves a deep knowledge about the manufacturing system and a good analysis of the system performances through KPI (Maier-Sperdelozzi and Hu, 2002).

The diagnosis should reveal the level at which the reconfiguration might be carried: machine level or system level for example (Capawa Fotsoh et al., 2020). Herein an analysis of the performances through KPIs at each level is required. This task is often difficult due to the complexity of the systems. One way to overcome this issue is to use modularity. (Lameche et al., 2017) states that modularity reduces the system complexity and allows the organization of the system in a set of distinct functional component that can be developed and studied independently. That is, the system is broken into number of subsystems each having its KPIs. As KPIs results from the behavior of the system, the analysis of the KPIs at a given time should take into account the control logic applied at that time.

Once the diagnosis is done and the reconfiguration point identified, a new configuration is chosen. This configuration has to be evaluated before its implementation.

Simulation is widely used to address this issue. (Cardin and Castagna, 2006), (Lateef-Ur-rehman, 2013). Yet, building a simulation model of a complex system, is difficult. Nevertheless, the use of the system modularity could lead to a modular build of the simulation model. That is, each subsystem is faithfully represented by a simulation model, as it is in real life. This principle of modularity in simulation models has been widely applied during the last decades to easily evaluate manufacturing system (Hibino et al., 2002).

Therefore, modularity is seen as an answer to the system complexity. It helps in a deep knowledge about the system and guide the reconfiguration strategy. As the reconfiguration decision is made regarding the system level, the evaluation of the results of each subsystem have to be considered at the system level. Thereby, arises the need to aggregate subsystem’s KPIs, simulation models and even control logic. This issue is the one addressed by this paper. Our target is to investigate the modularity of the system regarding the simulation model, control logic and KPI of the system, and to study the aggregation of the module in order to have accurate data at the system level for reconfiguration decision making. Far from proposing rigid models of aggregation, the aim of the paper is to provide opportunities of thinking about the notion of aggregation of simulation models, KPI and control logic within manufacturing context. Herein, the aim of the paper is to highlight the importance of aggregation, especially within the process of reconfiguration decision making for modular manufacturing systems.

The remainder of this article is as follow: section 2 presents the modular vision of RMS, section 3 states the

aggregation problem and present the role of aggregation. Section 4 uses an illustrative example to present some ideas about aggregation methods. Finally section 5 discusses and concludes the paper.

2 A MODULAR VISION OF RMS

Modularity helps in reducing the system complexity. It concerns physical and logical aspect (Koren and Shpitalni, 2010), but also KPIs and even simulation model used to represent the system. A module is then considered as a system element with four aspects: physical, logical, KPIs and simulation. The aggregation of several modules constitute the system. Hence, choosing a new configuration will be choosing the modules that will be put together (aggregated) and define how the aggregation will be made. The highest module represents the system itself.

The physical aspect of the module (PM) refers to tangible element in the real life. It could be a workstation, operators, storage system, etc. It is managed by a control logic, and has related KPI and simulation model. Before implementing the PM in real life, its simulation model can be used for evaluation.

The control aspect (CM) describes the behavior and manages the PM. It represents the intelligence of the PM. CM is faithfully replicate in the simulation aspect of the module, to ensure the correspondence between the real world and virtual world.

The simulation aspect (SM) is the virtual representation of the module. It allows to evaluate modules as it would be in real life situation. Simulation aspect of the module gives an insight of the system in case of reconfiguration process.

The KPIs measure the performance of the module. Value could either be measured on the real system or simulated. It allows the diagnosis of the module, and triggers the reconfiguration decision.

A representation of this module-based view of RMS is given on (Fig. 5). This module based view of the system allow to reduce the system complexity regarding the design or even the evaluation of the configuration, thus ease the reconfiguration process. The remaining issue concerns the aggregation of the modules

3 AGGREGATION: PROBLEM STATEMENT AND ROLE

3.1 Problem statement

To better understand the issue, let us consider the following example: A manufacturing system made of two workstations M1 and M2. M1 behaves following the FIFO (First in First out) rule and its utilization rate is represented by the curve denoted M1 on Fig. 1. Whereas M2 follows a particular control logic, depending on the product and its utilization curve is denoted M2 on Fig. 1.

In order to study the behavior of the system, the models of M1 and M2 are put together. It is therefore necessary to consider an aggregation function that will

allow the system (upper level) to combine the different information from the lower levels (i.e from M1 and M2). For example, what would be the utilization rate to consider? Fig. 2 gives an glimpse of possible answers. At the system level, solution 1 considers the utilization rate as the sum of M1 and M2. Solution 2 considers maximum (Max) between M1 and M2, and solution 3 the average between M1 and M2.

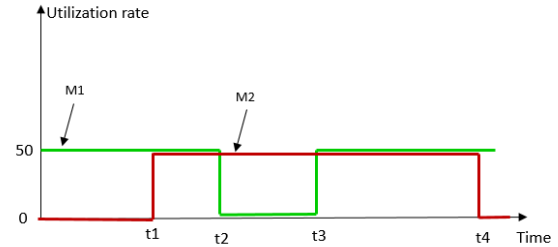


Fig. 1: utilization rate shape per workstations

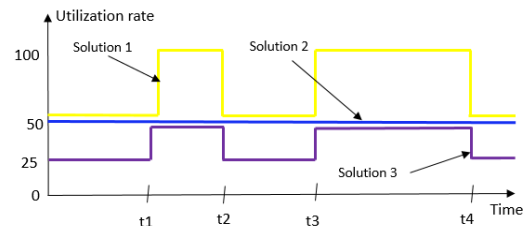


Fig. 2: utilization rate shape for the system

A simulation model of M1 is build, as well as for M2. Both faithfully represent the workstation and their behavior as it might be in real life situation, so that the study of the simulation model would help in reconfiguration decision. While building the system simulation model, what would be the system behavior, i.e. what would be the control logics at the system level?

3.2 Role of aggregation

Aggregation functions are usually defined and used to combine and resume several values into one, so that the end result of the aggregation takes into account, in a specified way, all the individual values. The notion of aggregation functions is widely used in statistics, economics, finance or even informatics (information retrieval) (Moulaoui, 2015), where aggregated values are often numerical or strings, as KPIs in manufacturing domain.

The reconfiguration decision at the system level takes into account the KPIs of each the module within a configuration. Even though the module is evaluated individually, it contributes to the system's performance. Therefore, the aggregation function of KPIs is needed and should be chosen carefully. Many mathematical functions are used to address numerical aggregation issue. Depending on the context and the purpose of the study, functions such as quadratic mean, minimum, maximum, standard deviation, etc. could be chosen (Bouyssou et al., 2006). As modules of the system have an impact on each other and exchange information. Aggregating two modules results in determining what will be KPIs and the behavior of the system. That is to de-

termine the control logic that will be applied at the system level to manage several aggregated modules. Herein, the aggregation of control logic is required (Leitão, 2004), as well as the aggregation of simulation model for a relevant evaluation of the system.

For simulation models, High Level Architecture (HLA) is a standard that promotes aggregation, interoperability between simulations, and aid the reuse of models in different context (IEEE 1516, 2000). More often, HLA is used when simulations use different several activities: each activity is represented by a simulation model which is then connected and coordinated with other model. HLA has proposed standards to support the interoperability within simulation of a whole system that involved several manufacturing actives such as materials activity, production activity and even commercial-marketing activity. Each activity often has its simulator, and the drawback that remains when applying the standards in real life is about combining models from different simulators (Pedrielli et al., 2011). The literature review conducted by (Thein, 2019) gives a great overview about previous works carried in this field. So far, the concept of HLA has been implemented for systems with different activities and simulations. The scope of this paper is limited on the production activity, where the simulation models of each sub model is built with the same simulator. Unlike HLA where the system will need a RTI (Run Time Infrastructure) to support information exchange between models, here the models belong to the same simulation software. It is therefore not necessary to transit information through a RTI. However, the different models must be able to communicate in a consistent manner to ensure the consistency of the production system model. Therefore, in addition to KPIs and control logic aggregation, simulation models have to be aggregated (Fig. 3).



Fig. 3: Role of the aggregation functions

4 ILLUSTRATIVE EXAMPLE

Let us consider a manufacturing line made of three workstations: M1, M2 and M3. The system manufactures two types of product (P1 and P2) that both uses M1 and M2 with different operational time here denoted time unit (tu). M3 is the same as M2. M1 behaves in FIFO, M2 gives priority to product P1 and M3 to product P2. There is a storage zone for M2 and M3. (Fig. 4).

The modular decomposition of the line considers the following modules: Et (entry), M1, M2 M3, Ex (exit). Each has its simulation model, KPI and logical control. The system model is obtained by putting together the module models. The KPI of the system depend on the

module’s KPI, and the behavior on a system level is added to manage control logic of each workstation. The following details the aggregation methods used for this example. FlexSim© is the simulation software used for this example.

Table 1: operation time by product by workstation

Product	Operation 1 on M1	Operation 2 on M2 or M3
P1	2 tu	5 tu
P2	2 tu	10tu

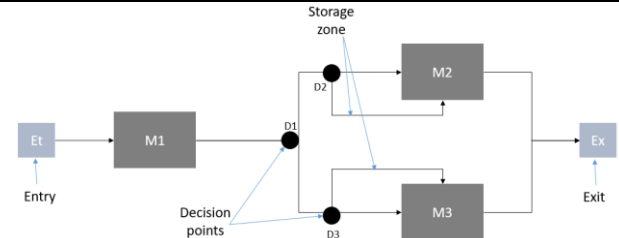


Fig. 4: system scheme

4.1 Aggregation of simulation models and control logic

For the aggregation of models, we propose to build simulation libraries. For each module, an object and the behavior describing the module are associated. The resulting element is stored in a library, so that whenever the module is needed, it can be easily extracted from the library. Thus, to each module of the system, we will associate a virtual module that describes it. The construction of the simulation model of the system will then import from the library all the modules we need. This saves time in the construction of the model, since the control logic relative to each module will already be associated to the module. For this example, Fig. 6 and Fig. 8 show the elements of the library created and the logic associated with each module.

Yet, the issue remains the communication between the elements of the library. To make them work together, a control logic that will trigger and manage the system is needed. We find here the notion of aggregations for the control logics. To manage the system very often the control logic at the system level is obtained by assigning priority order to the decisions taken by the different modules at specific times (Barbosa et al., 2015). These priorities can change over time, or remain fixed. They are very often logics of the type {if, ... else}.

In our case, the module models are retrieved from the library and a coordination logic is added. This logic may be stored in a library or not. It can be derived from priority rules as shown in this example. The priority order between M2 and M3 is manage by a system logic that switches between satisfying control logic of M2 and M3 or impose a new rule in order to avoid the system from stuck. In order words the logic at the system level is the following:

When workstations are available (i.e idle or there is a place in the storage), follow the priority of M2 and M3 (i.e send P1 on M2 and P2 on M3).
If M2(3) is busy (i.e its storage full) ask for the availa-

bility of M3(2). If M3(2) is available, send products (P1 and P2) on M3 (2), else, stop production on M1, till a workstation become available.

This rule is described by the logic on (Fig. 7). It is important to notice that, the logic both of the modules and the system, is built independently of the “object” representing the physical aspect. Thus we can easily add a new workstation (eg. Workstation type 1) to the system without deep changes. In other words, adding a new workstation means to add an object workstation from the library and associate the appropriate logic (the one describing the type 1 workstations), then add the behavior of the new workstation the system control.

The simulation model aggregation is possible with the use of a library, and the control logic aggregation, by defining an upper control that uses the lower control in addition to a priority rule.

4.2 Aggregation of KPI

Many KPI could be used for a system evaluation. (ISO 22400, 2014) described a set of KPI for manufacturing context. (Zhu *et al.*, 2018) proposed a framework to organizing KPI regarding three types : measurement, equipment and process. For our example, we choose to study aggregation of one KPI from each section: Work In Progress (WIP) for measurement, utilization rate (U) for equipment and Throughput (T) for process.

Simulated valued for modules are given in **Table 2**.

4.2.1 Aggregation of WIP

At the system level, the WIP is considered as the sum of the WIP of each workstation. Of course the workstations include possibilities for storing product awaiting production.

$$WIP_{System} = \sum_{i=1}^n WIP_{Mi} + \epsilon_{WIP} \quad (1)$$

$$WIP_{System} = WIP_{M1} + WIP_{M2} + WIP_{M3} + \epsilon_{WIP} \quad (2)$$

ϵ_{wip} represents the products located on the transportation equipment between two successive work stations. If the two workstations follow each other (without transport agent) then ϵ_{wip} is zero. In some applications ϵ_{wip} can be neglected. However, for a complete analysis and evaluation at system level, it is better to take it into account. The use of this aggregation function allows us to determine the system WIP, and thus to evaluate the performance of the configuration. A reconfiguration decision could aim at reducing the system WIP.

4.2.2 Aggregation of Utilization rate

The simulation shows that M2 and M3 are almost completely used; whereas, M1 is still available up to 40%. The utilization rate of the system should integrate this two data. More often, the utilization rate at the system level is calculated by using the arithmetic mean of workstation. This could be explained by the fact that workstations are often the most requested in the system. In

this study we realize that the result obtained by just considering the workstation gives a glimpse, but not a very representative state of the system. Hence, we propose to considered the utilization rate of other modules such as transportation, storage. The aggregation will be made as follow:

$$U_{System} = \frac{\sum_{i=1}^m U_{Mi} + \sum_j^{n-m} U_{Other\ than\ Mi}}{n} \quad (3)$$

$$U_{System} = \frac{U_{M1} + U_{M2} + U_{M3} + U_{Et} + U_{Ex}}{5} \quad (4)$$

4.2.3 Aggregation of Throughput

The throughput represents the average number of products that can be manufactured per hour. The simulation for our example shows that, M1 has a higher throughput than M2 and M3. This can be explained by the fact that, the product coming from M1 are split between M2 and M3 for final operations. Thus, at the system level, the throughput will depend both on the throughput of M1 and on the sum of the throughput of the workstations that perform the last operation of the task list (M2 and M3). The throughput at the system level is then obtained by considering the minimum between these two:

$$T_{System} = Min \{ (T_{M2} + T_{M3}); T_{M1} \} \quad (5)$$

5 DISCUSSION AND CONCLUSION

The aggregation method for simulation and control logic models is implemented by using libraries containing representations of each module of the system. The construction of a new configuration consists of importing the necessary objects from the library, associating the necessary control logic to them, and finally, defining the logic to be applied to the whole system, according to the needs and objectives of the configuration. With the modular construction of the simulation model, the proposition of a new configuration can be easily tested and evaluated. This significantly reduces the time required to build the simulation model. Nevertheless, the concept of the simulation library remains a solution that can be adapted to various manufacturing contexts, as long as the system is modular.

About KPIs, their analysis allows an in-depth diagnosis of the system and allow to determine the causes of possible disturbances. For our example, we used simulation to obtain KPIs of each module, nevertheless, these valued could be measured on the real system. KPIs can lead to the identification of a bottleneck station in the system and thus lead to a reconfiguration decision. E.g. The system is too much busy because M3 is full (U =99%) and its Storage always full (WIP=10), means that M3 is a bottleneck. A reconfiguration decision could be to duplicate M3. As the system and the KPIs are modular, it is easier to identify the origin of the problems and to provide a targeted solution. However, the KPI aggregation functions proposed here remain

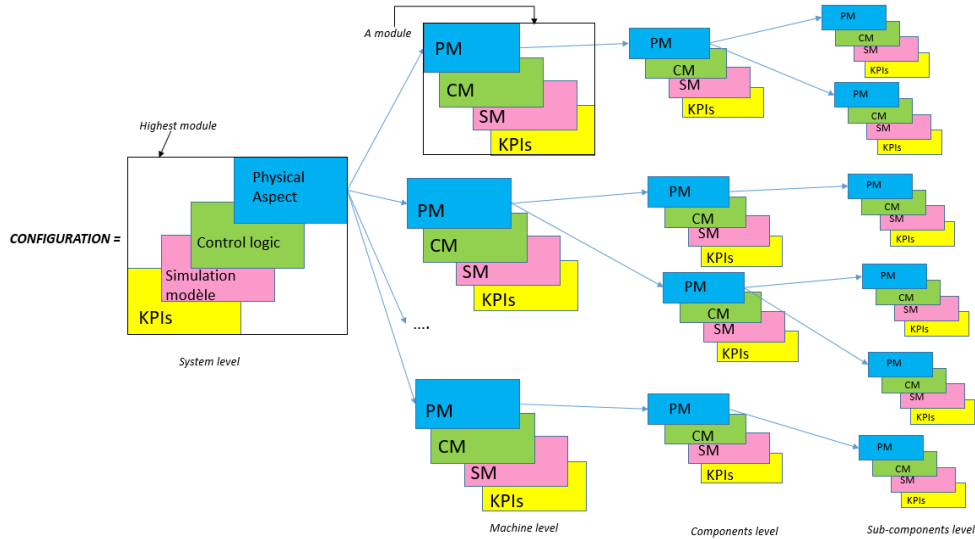


Fig. 5: A modular vision of the system

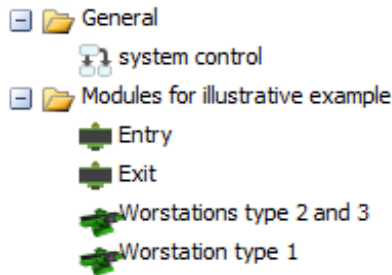


Fig. 6: Library of modules

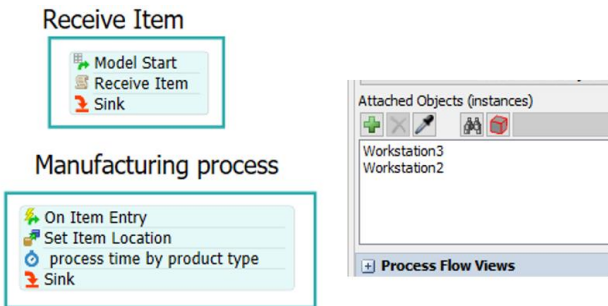


Fig. 8: control logic with attached workstations

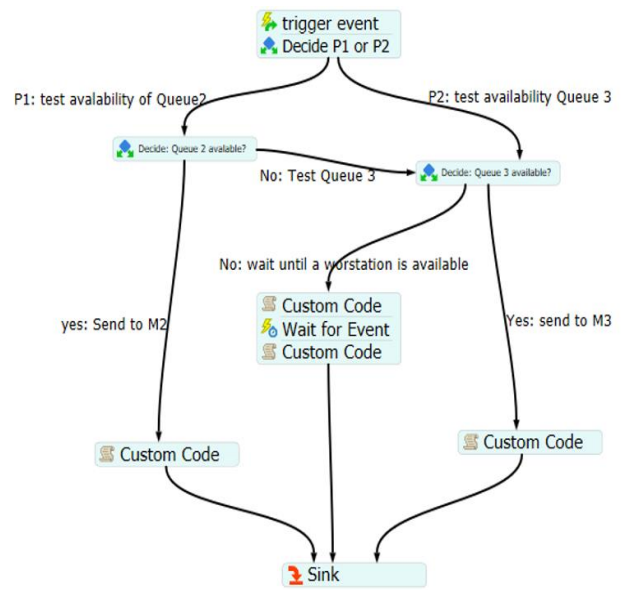


Fig. 7: control logic that manages the system

Table 2: Modular analysis of KPIs

	Machine level					System level
	ET	M1	M2	M3	Ex	Configuration (aggregation)
U(%)	99.79	53.55	99.86	99.92	99.85	90.6
T	966	965	507	454	960	960
WIP	--	9	10	10	--	29

specific to the example. But, they can be extended to similar systems (i.e. modular), or inspire the definition of aggregation functions for other modular systems.

Aggregation is important for a relevant analysis and evaluation of the system, and the notion of modularity helps by reducing the complexity and guiding to a relevant diagnosis of the system. Yet, modularity could be a source of complexity in the analysis if the relationships between modules are not well defined. As production contexts are very different, it is very difficult to generalize aggregation functions. But the idea of building simulation libraries and control logics already allows a step forward in the application of modularity. The construction of new configurations by module aggregation is similar to the formation of holarchies in holonic systems (Chacón *et al.*, 2012) and to the modular construction of digital twins (Redelinghuys *et al.*, 2019). The notion of aggregation proposed here can thus be used to feed some discussion concerning the issue about aggregating data and models within holonic system or digital twins.

ACKNOLEGDMENT

This research work is supported by funding of the PhD program PERFORM (Fundamental research and development program resourcing on manufacturing) from the IRT Jules Verne (<https://www.irt-jules-verne.fr/>).

REFERENCES

- Barbosa, J. et al. (2015) 'Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution', *Computers in Industry*, 66, pp. 99–111. doi: 10.1016/j.compind.2014.10.011.
- Bouyssou, D. et al. (2006) *Concepts et méthodes pour l'aide à la décision*, volume 1, outils de modélisation. Hermès. Available at: <http://www.eyrolles.com/Informatique/Livre/9782746212893/livre-concepts-et-methodes-pour-l-aide-a-la-decision-volume-1.php?xd=aaf52345a1d2961020198284312840ab>.
- Capawa Fotsoh, E. et al. (2020) 'A Classification for Reconfigurable Manufacturing Systems', in Benyoucef, L. (ed.) *Reconfigurable Manufacturing Systems: From Design to Implementation*. Cham: Springer International Publishing, pp. 11–28. doi: 10.1007/978-3-030-28782-5_2.
- Cardin, O. and Castagna, P. (2006) 'Handling uncertainty in production activity control using proactive simulation', *IFAC Proceedings Volumes (IFAC-PapersOnline)*. IFAC, 12(PART 1). doi: 10.3182/20060517-3-fr-2903.00297.
- Chacón, E. et al. (2012) 'Planification en ligne pour les systèmes de production distribués: une approche par les systèmes holoniques'.
- Hibino, H. et al. (2002) 'Manufacturing adapter of distributed simulation systems using HLA', in, pp. 1220–1228.
- IEEE 1516 (2000) *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)*.
- ISO 22400 (2014) *Automation Systems and integration - Key performance indicators (KPIs) for manufacturing operations management*.
- Koren, Y. and Shpitalni, M. (2010) 'Design of reconfigurable manufacturing systems', *Journal of Manufacturing Systems*. Elsevier Ltd, 29(4), pp. 130–141. doi: 10.1016/j.jmsy.2011.01.001.
- Lameche, K. et al. (2017) 'Modularity in the design of reconfigurable manufacturing systems', *IFAC-PapersOnLine*. Elsevier B.V., 50(1), pp. 3511–3516. doi: 10.1016/j.ifacol.2017.08.939.
- Lateef-Ur-rehman, A. (2013) 'Manufacturing configuration selection using multicriteria decision tool', *International Journal of Advanced Manufacturing Technology*, 65(5–8), pp. 625–639. doi: 10.1007/s00170-012-4201-5.
- Leitão, P. (2004) *An Agile and Adaptive Holonic Architecture for Manufacturing Control*. University of Porto.
- Maier-Speredelozzi, V. and Hu, S. J. (2002) 'Selecting manufacturing system configurations based on performance using AHP', *Technical Paper Society of Manufacturing Engineers MS*, (MS02-179), pp. 1–8. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0-0036300907&partnerID=40&md5=9c626532220edfae6e8665340a066dcd>.
- Mehrabi, M. G., Ulsoy, A. G. and Koren, Y. (2000) 'Reconfiguration manufacturing systems: Key to future manufacturing', *Journal of intelligent manufacturing*, 11, pp. 403–419.
- Moulahi, B. (2015) *Définition et évaluation de modèles d'agrégation pour l'estimation de la pertinence multidimensionnelle en recherche d'information*. Available at: <http://thesesups.univ-tlse.fr/2977/%0Ainternal-pdf://0.0.0.70/2977.html>.
- Pedrielli, G. et al. (2011) 'Simulation of complex manufacturing systems via HLA-based infrastructure', in *Proceedings - Workshop on Principles of Advanced and Distributed Simulation, PADS*. IEEE. doi: 10.1109/PADS.2011.5936772.
- Redelinghuys, A., Basson, A. and Kruger, K. (2019) 'A Six-Layer Digital Twin Architecture for a Manufacturing Cell', in *Service Orientation in Holonic and Multi-Agent Manufacturing*, pp. 273–284. doi: 10.1007/978-3-030-03003-2.
- Thein, N. N. M. (2019) 'Implementing Data Distribution Management System Using Layer Partition-based Matching Algorithm', *Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering, WCSE 2019 SPRING*, 9(1), pp. 68–72. doi: 10.18178/wcse.2019.03.012.
- Zhu, L. et al. (2018) 'Key performance indicators for manufacturing operations management in the process industry', *IEEE International Conference on Industrial Engineering and Engineering Management, 2017-December(December)*, pp. 969–973. doi: 10.1109/IEEM.2017.8290036