



HAL
open science

Hybrid production-system control-architecture for smart manufacturing

Michele Dassisti, Antonio Giovannini, Pasquale Merla, Michela Chimienti,
Hervé Panetto

► **To cite this version:**

Michele Dassisti, Antonio Giovannini, Pasquale Merla, Michela Chimienti, Hervé Panetto. Hybrid production-system control-architecture for smart manufacturing. 12th OTM/IFIP Enterprise Integration, Interoperability and Networking Workshop (EI2N), Oct 2017, Kallithea, Rhodes, Greece. pp.5-15, 10.1007/978-3-319-73805-5_1. hal-01793629

HAL Id: hal-01793629

<https://hal.science/hal-01793629>

Submitted on 16 May 2018

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.

Hybrid Production-System Control-Architecture for Smart Manufacturing

Michele DASSISTI¹, Antonio Giovannini²
Pasquale Merla³, Michela Chimienti² and Hervé Panetto⁴

¹Politecnico di Bari, DMMM, Bari, Italy `michele.dassisti@poliba.it`

²INRESLAB Scarl, Monopoli (BA), Italy (`a.giovannini,`
`m.chimienti`) `inreslab.org`

³Ali6 Srl, Monopoli (BA), Italy `merla@ali6.it`

⁴CRAN, University of Lorraine, CNRS `Herve.Panetto@univ-lorraine.fr`

Abstract. Highly customized products with shorter life cycles characterize the market today: the smart manufacturing paradigm can answer these needs. In this latter production system context, the interaction between production resources (PRs) can be swiftly adapted to meet both the variety of customers' needs and the optimization goals. In the scientific literature, several architectural configurations have been devised so far to this aim, namely: hierarchical, heterarchical or hybrid. Whether the hierarchical and heterarchical architectures provide respectively low reactivity and a reduced vision of the optimization opportunities at production system level, the hybrid architectures can mitigate the limit of both the previous architectures. However, no hybrid architecture can ensure all PRs are aware of how orienting their behavior to achieve the optimization goal of the manufacturing system with a minimal computational effort. In this paper, a new "hybrid architecture" is proposed to meet this goal. At each order entry, this architecture allows the PRs to be dynamically grouped. Each group has a supervisor, i.e. the optimizer, that has the responsibility: 1) to monitor the tasks on all the resources, 2) to compute the optimal manufacturing parameters and 3) to provide the optimization results to the resources of the group. A software prototype was developed to test the new architecture design in a simulated flow-shop and in a simplified job shop production.

Keywords: Cyber Physical Production System; Factory automation; Hybrid Control Architecture; Intelligent Manufacturing System; Reconfigurable Manufacturing System.

1 Introduction

Market mutations from a local to a global economy are urging production systems to face new challenges. Recent years have witnessed the switch from mass production to mass customization, with customers requiring product even more customized, maintaining high quality and low prices [1], [2]. Manufacturing companies, to remain competitive, should improve their ability to swiftly adapt their production capacities to this demand for variety. In a smart manufacturing scenario [3], [4], the production control-

system architectures are strategic as a orchestrating system of manufacturing parameters at each production resource (PR).

Several scientific papers in recent years proposed solution to these challenges by devising distinct types of control architectures (CA). There are three main approaches that correspond to three main architecture of production system control classes [5]: Fully Hlerarchical architectures (FHI), Fully HETERarchical architectures (FHE) and Hybrid Control Architectures (HCA). The centralization of the FHI reduces the reactivity to unpredictable events. The decentralization of the FHE reduces the optimization opportunities.

The current HCA do not ensure all the production resources are always aware of the operating parameters to meet the optima conditions. In this paper, we propose a new HCA to provide a “global view”, i.e. the resources involved in the manufacturing process should have information about how to behave to achieve the optimization goal of the manufacturing system. Moreover, to deal with complex manufacturing systems characterized by a high production variety, attention is paid to the computational effort required to ensure the global view in the proposed CA.

In our proposal, each production resource (PR, made by a physical part, e.g. the operating machine, and a logical part, e.g. control logic) is aware of the optimal behavior (i.e. set of manufacturing parameters) that that PR can apply when no unpredictable events should be managed locally, e.g. a delay of the supplier PRs.

The paper is organized as follow. Section 2 states the problem and the operating scenario for the proposed architecture and shows as the existing CAs would face the problem. Section 3 details the proposed architecture and explains how it face the stated problem. Section 4 shows the software prototype. Section 5 explains the tests and the validation of the architecture. In the Section 6, the conclusions and the future perspectives are discussed.

2 Literature review

2.1 Problem Statement

As identified above, this paper aims to design and develop a CA for production systems, characterized by a high production variety, therefore this CA, also considering the challenges proposed in [6] by IFAC TC 5.3, should meet the following requirements:

- all PRs must have a constant availability (i.e. with a high refresh rate) of the global view, i.e. each PR should have the availability of the optimal parameters setting related to the best available optimization goal;
- the CA must be reactive with a low computational effort.

2.2 Control architectures

This section presents a review of the production CAs in the scientific literature. The papers are discussed according to the classification in [5] that identifies three main classes of CAs:

- FHI: the PRs presents in the production system have master/slave interactions;
- FHE: the PRs have the maximum decisional autonomy;
- HCA: this class includes this architectures combine the advantages of both the previous classes, switching from hierarchical to non-hierarchical and vice versa.

2.3 Architecture type

The FHI architectures can partially satisfy the problem statement but, due to their structural characteristics, these are rigid and do not guarantee the reactivity required to control a high production variability. If we apply this architecture type to the problem proposed, it results that all knowledge and decisional ability it will be given to the Central Unit (Fig. 1(a)). Since all the computational effort resides on the Central Unit, it is not possible to guarantee the required reactivity: the time necessary to estimate the optimal parameters settings for all devices (according to global optimization rule) as well as the central-unit effort required to transfer all the information to all the devices under its control increases the basis of the number of resources. If one add intermediate units to control each cell (understood as a group of PRs i.e. a production department) that do not interact with each other (e.g. [7]), the computational burden to get higher refresh rates lowers, but there is a less complete global view (Fig. 1 (b)).

On the opposite, the FHE architectures (e.g. [8]) ensure the continuous reconfiguration of the PRs that have learning and decisional abilities. These architectures are not able to guarantee always the global view, since they are affected by “short-sightedness” due to the lack of a central authority that suggests the best behavior to achieve the global optimization. This latter is often in conflict with the local objective of the machines. If one adopts a heterarchical architecture (Fig. 2), each PRs (physical part or software component) can take decisions and act behaviors to reach a local goal and to interact with each other. In this way, no resource controller can provide the global view to each other because each try to achieve its local goal.

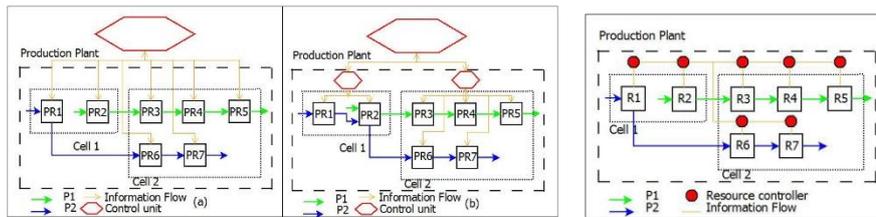


Fig. 1. Scheme of a Hierarchical architecture **Fig. 2.** Scheme of an Heterarchical architecture

The following sections analyze the HCA, that aim to combines the advantages of both the other two classes.

2.4 Hybrid Control Architectures

The HCA [9]–[19] can be classified according to the degree of evolution of the control structure during the execution of production [11]: architectures can be dynamic (D-

HCA) or static (S-HCA). They differ because of the evolution of the control system, which evolves in D-HCA and remains the same in S-HCA. As described in Section 2, it is mandatory that in all operating situations, the global view of the system and their optimal manufacturing parameter settings (e.g. temperature, velocity) is available for all the production resources to achieve the global optimization.

2.5 S-HCA

In S-HCA[14]–[18] the control does not evolve, i.e. the configuration is the same all of time. They are very close to the FHI architectures, but in this case, the lower levels are always can refuse the guidelines to the upper level, e.g. they can ask to upper level make a new reconfiguration. Apparently, these architectures can meet completely the problem statement and for this, each of them has been analyzed in detail.

In [17] a monolithic structure is adopted which does not allows flexibility (see case in Fig. 1(b)): here only one controller for each production cell is adopted that do not interact with the others. Product P1 must pass in both the cells and since the interaction between cells is lacking, the optimization will concern only cell 1.

When resource R2 releases P1 the optimization will concern only cell 2 and so on. In [18] the requests to merge the individual interests is triggered directly from the PRs that as in case of FHE do not have the global view of the system. In [15] instead a dynamic group definition is proposed, although how it forms the cluster is unclear. The major issue is that the PRs communicates only with the mediators and there may be bottlenecks if the number of resources increases, as in FHI. In [14] two different agents have been used to improve the production: Order (OA) that contains the production plan for a single production order and Resource (RA) that corresponds to a single production resource. When a new order arrives, the OA verifies its feasibility and after questioning the RA, according to their responses, the OA inserts the new activities in the Gantt of each RA. The OA monitors the production advancement. The RA managed their own Gantt. They react to little perturbation changing slightly and if a major perturbation occurs, the RA that has detected the perturbation, asks to corresponding OA to reschedule. In this case, each OA limits to allocate the resources and it make a re-scheduling only if a RA so requests. In this way is not possible ensure the global view because the RAs managed their Gantt and the re-orchestration request trigger from themselves. Only if the OA is continuously to the research of the best possible solution, rather than to be only in wait, this architecture could be completely satisfy the problem statement. In [16] defined hierarchy is adopted based on the assumption that changes in production plans are infrequent. In this way, the intermediate controllers are defined during the design of the system and they do not interact with them. For example, let us apply this architecture to the scenario shown in Fig. 3 where there are 9 resources, 3 products and 3 production Cells. The result will be an upper level that monitor all production resources and an intermediate control level composed by a control unit for each production cell. Each intermediate control unit tries to achieve the optimum for its own cell, based on its own goals, which can be in contrast with each other.

2.6 D-HCA

The D-HCA [9]–[13] cannot satisfy completely the sustainability of production control. These in fact start with a hierarchical configuration, but under a disruption, the PRs do not respect the “authority” of their control unit: the control system thus evolves from hierarchical to heterarchical. Some situations are possible in which the production resources operate as fully independent entities (heterarchical configuration), and thus the constraint posed in the problem statement will be only partially fulfilled, since in the heterarchical configuration is not possible to ensure the global view.

Finally, a comparison is required with PROSA [19], where exists a software supervisor (Staff Holon (SH)) that represents an external expert that suggests advices to PR (PRs can refuse advices). However, the amount of SHs and their knowledge should be fixed a priori. In a high production variety scenario, this constraint and the unpredictability of the production can result in an under- or over-estimation of the amount of the required computational effort and knowledge resources to implement the SHs.

The next section proposes a new architecture, where groups and supervisors are dynamically defined and dynamic is also the knowledge assignment to the supervisor.

3 The new “hybrid architecture”

From the analysis of these works emerges that the existing CAs only partially fulfill the requirements to solve the proposed problem statement. The architecture here proposed belongs to the HCA class that fulfils the requirement of the smart manufacturing paradigm.

A central unit configures groups of PRs based on each order entry. Each group is independent, i.e. each product (P) requires the resources of only one group of PRs to be finalized. The central unit defines an optimizer O for each group of PRs. This O 1) gathers periodically the states of the PRs in the groups 2) compute the optimal process parameters for all the PRs in the group and 3) provide to the PRs the results of the optimization.

To preserve the global view with minimal effort, and thus assure adequate flexibility, our architecture tries to solve the following problems:

- to dynamically create (re-orchestration) the intermediate control units (Optimizer O) when the production plan change (a new order is provided), to decide, case by case, the PRs under their control (Optimization Group OG) and to create dynamically the communication channels between the PRs belonging to the same OG;
- to provide only to Os, dynamically defined by the CU, only the necessary knowledge according to the optimization goal of the manufacturing system. Doing so, the computational effort is dynamically transferred from the CU to the Os.

Both these features make the proposed architecture prone to be implemented in a smart manufacturing context. The next sections detail the behavior of the proposed architecture both in re-configuration and in execution phase.

3.1 Re-configuration phase

The CU starts from a waiting state for an introduction of a new order (PO).

- Step 1: the CU receives a PO composed by products set that are characterized from their working sequence (operations, e.g. lamination, cooking etc. and PRs linked to them);
- Step 2: the CU sends a stop signal to all available PRs and asks them their advancement state (the completed operations until that moment without the ongoing operations);
- Step 3: the PRs send the answer to the CU. If they are running a process they continue it, otherwise stop;
- Step 4: the CU when receives all answers creates the new production plan merging the new PO with the operations of the old production plan that the PRs must still perform, if they exist;
- Step 5: the CU starting from the new production plan, identifies the independent paths (a path is defined as the machines sequence that allows passing from raw material to finished product and two paths are independents if they not cross i.e. if they do not share no PR). Each path corresponds to an OG;
- Step 6: for each OG, a PR (that for as they were defined paths, the PRs can belong to only one OG at a time) is chosen to assume the role of O. It will have under its control and will be able to interact only with the PRs (slaves) belonging to its OG. The difference between the slaves and the Os resides in an executable software that the CU compiles and sends to each PRs in this step, according to their role;
- Step 7: the CU sends to each PR its own new production plan (agenda) and it relies on the Os to manage their own OG;
- Step 8: the CU returns to waiting for a new PO to make a new re-orchestration.

The Fig.4(a) shows the condition of a production plant during the execution phase (three OG have already been defined), instead the Fig.4(b) shows the result of a re-configuration when a new PO (it contains P5 and P6) is introduced. After the re-configuration, the OG (and then the O's) have become two and they are different from those in the above condition.

After the re-configuration, the production can restart. Step 2 highlights how the PRs send only the finalized operations without considering the ones in progress. The ongoing operations are considered in the new agenda: when the production resumes, their duration is updated based on the percentage of work done during the re-configuration stage.

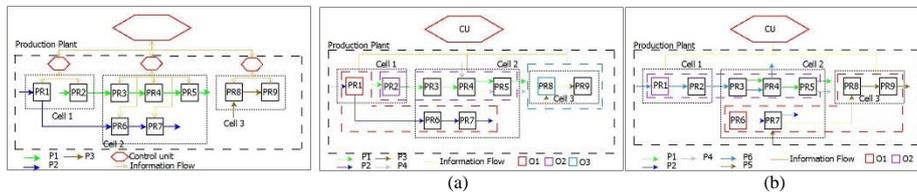


Fig. 3. Application of [16]

Fig. 4. Re-configuration phase

3.2 Execution phase

The execution phase starts after each PR receives its executable software and its updated agenda that contain all products that it must perform in a defined sequence. During execution phase (the fluxes between PRs are shown in Fig. 5):

1. each O reads the state (the value of the controllable manufacturing parameters to achieve an optimization result, e.g. velocity) of each PRs under its control;
2. each O sends to all PRs belonging its OG the best manufacturing parameters setting to achieve a global goal according to an optimization rule (e.g. energy saving);
3. each PRs, including Os, carry on the production expected from their agenda, setting their manufacturing parameters as suggested from their O, if it is feasible and they can react quickly to unpredictable events.

As described above, a fundamental characteristic of this architecture is that thanks to the dynamic creation of the OG, any global optimization problem is divided in more small problems, allowing to maintain the global view and minimize the computational effort. In this way, an intermediate control unit manages fewer PRs than those managed by the CU, with a consequent performance boost that affect the time required to have an appropriate solution by an optimization algorithm (Heuristic, Genetic Algorithm etc.) as well as the reaction time.

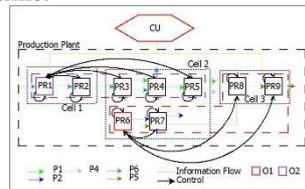


Fig. 5. Execution phase

This work does not consider the PRs' breakdown, allocation or re-allocation.

In the next sections, an example to validate this architecture is proposed. It will be demonstrated how this architecture can satisfy completely the problem statement.

4 Software prototype

To test if the dynamic re-configuration of the intermediate control level is possible with the architecture proposed, a software prototype was implemented. The focus was on the global view. To test the dynamic grouping, the reference scenario adopted for the test is composed by 4 PRs (PR1, PR2, PR3, PR4).

To simulate the CU and the PRs it has been used a Linux Debian terminal for each entity. As described in Section 3, the POs contain the products set with their working sequence characterized from operations and PRs linked to them and we assume that the PO is correct, i.e. there is not the possibility that a PO cannot be performed. The PRs presentation board and the PO was implemented in XML (eXtensible Markup Language).

The behavior of the CU and those of the PRs (the code to generate the executable program in Section 3) have been implemented in C++. The PRs' production plans (agendas) are text files. The optimization goal is the energy saving and the algorithm to find the best manufacturing parameter setting is a heuristic. To allow the communication between CU and PRs, TCP/IP protocol has been used.

5 Tests

This section describes the tests that allow to show how the proposed architecture fulfill the problem statement. Two PO have been inserted. The first PO has two products, P1 (to be machined on PR1, PR2, PR4) and P2 (to be machined on PR4). The second one has two products, P3 (to be machined on PR2, PR4) and P4 (to be machined on PR2, PR3, PR4).

When PO1 is sent, the CU defines an OG formed only by PR1, PR2, and PR4, since there are not products that involve PR3. Then, the CU assigns to PR1 the O role.

The Fig. 6(a) shows the agenda of all PRs involved in the optimization. It is possible to notice that: 1) only the PR1 agenda contains information related to all products and all PRs belonging to its OG (in red circle); 2) PR2 and PR4 agendas contain information related to O IP (blue circle), IP of their previous PR, and operations that they must perform (green circle).

PR1 completed all planned operations and then PO2 starts working. The CU performs a re-configuration: as shown in Fig. 6 (b), the OG dynamically changes (now is composed only by PR2; PR4; PR3) and PR2 is chosen as O.

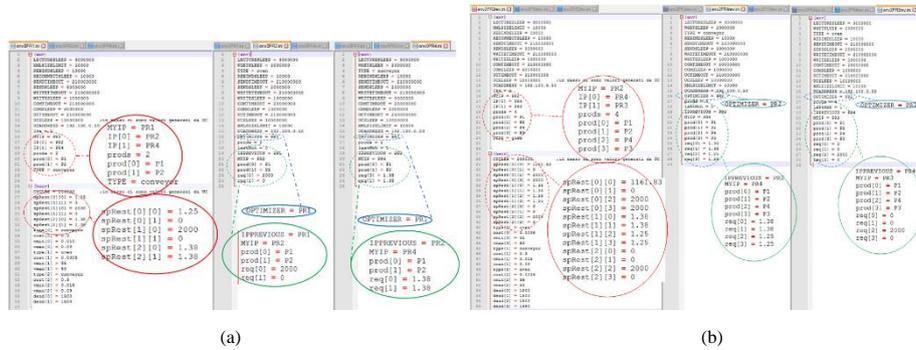


Fig. 6. (a) Agenda of PRs belonging to OG after PO1 insertion (b) Agenda of PRs belonging to OG after PO2 insertion

Fig. 7 shows on left side an example of an agenda that Os sends to a PR belonging to its own OG. Each row in Fig. 7 is structured as follows: on left side <starting time>;<ending time>;<parameter to apply> and on right side <reading time>;<value read>. Moreover, Fig. 7 shows on right side the lectures of the same PR in the same period and shows the PRs refusing the suggested parameters if a product is on delay: PR readings are 0 on the highlighted area, despite the values in the plan, since a product is on delay.

6 Conclusions: limits and future perspectives

In this work, a new hybrid control-system architecture is presented. As discussed in section 2, no-architecture in the literature could meet the requirements in the problem statement, since the FHI architectures cannot ensure the necessary reactivity, the FHE architectures cannot ensure the global optimum, and HCA cannot ensure both the requirements of global view with the required reactivity. By means of the tests described in Section 5, it is possible to show that the new proposed HCA fulfill the problem statement:

- All PRs have the constant (i.e. with a high refresh rate) availability of their own best manufacturing parameters, necessary to ensure the global view;
- By means of the dynamic grouping and of the dynamic assignment of the knowledge, the computational effort is re-allocated at each order entry.

The limits of the proposed architecture concern the limit in sharing products between different PR groups: this point makes the proposal less effective in a general job-shop scenario. Consequently, a future perspective can be the extension of the application domain to all job shop cases, assembly systems and the handling of the resource allocation.



Fig. 7. Predicted (left) and real (right) production plan

Acknowledgement

This research was partially supported by Gestal2000 Srl which committed to InResLab Scarl the project "GEM - Gestal Energy Management".

References

- [1] M. Dassisti and M. D. Nicolò, "Enterprise Integration and Economical Crisis for Mass Craftsmanship: A Case Study of an Italian Furniture Company," in *On the Move to Meaningful Internet Systems: OTM 2012 Workshops*, P. Herrero, H. Panetto, R. Meersman, and T. Dillon, Eds. Springer Berlin Heidelberg, 2012, pp. 113–123.

- [2] M. Brettel, N. Friederichsen, M. Keller, and M. Rosenberg, "How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0 perspective," *Int. J. Mech. Ind. Sci. Eng.*, vol. 8, no. 1, pp. 37–44, 2014.
- [3] Kagermann et al., "Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group," 2013. .
- [4] M. Dassisti *et al.*, "Industry 4.0 paradigm: The viewpoint of the small and medium enterprises," in *7th International Conference on Information Society and Technology, ICIST 2017*, Kopaonik, Serbia, 2017, vol. 1, pp. 50–54.
- [5] D. Trentesaux, "Distributed control of production systems," *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 971–978, Oct. 2009.
- [6] H. Panetto, M. Zdravković, R. Jardim-Goncalves, D. Romero, J. Cecil, and I. Mezgár, "New Perspectives for the Future Interoperable Enterprise Systems," *Comput. Ind.*, vol. 79, pp. 47–63, Jun. 2016.
- [7] J. Pfrommer, D. Stogl, K. Aleksandrov, N. S. Escaida, B. Hein, and J. Beyerer, "Plug & produce by modeling skills and service-oriented orchestration of reconfigurable manufacturing systems," - *Autom.*, vol. 63, no. 10, pp. 790–800, 2015.
- [8] S. Weyer, M. Schmitt, M. Ohmer, and D. Gorecky, "Towards Industry 4.0-Standardization as the crucial challenge for highly modular, multi-vendor production systems," *IFAC-Pap.*, vol. 48, no. 3, pp. 579–584, 2015.
- [9] P. Leitão and F. Restivo, "ADACOR: A holonic architecture for agile and adaptive manufacturing control," *Comput. Ind.*, vol. 57, no. 2, pp. 121–130, Feb. 2006.
- [10] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, "Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution," *Comput. Ind.*, vol. 66, pp. 99–111, Jan. 2015.
- [11] C. Pach, T. Berger, T. Bonte, and D. Trentesaux, "ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling," *Comput. Ind.*, vol. 65, no. 4, pp. 706–720, May 2014.
- [12] J.-F. Jimenez, A. Bekrar, G. Zambrano-Rey, D. Trentesaux, and P. Leitão, "Pollux: a dynamic hybrid control architecture for flexible job shop systems," *Int. J. Prod. Res.*, pp. 1–19, Aug. 2016.
- [13] G. Zambrano Rey, C. Pach, N. Aissani, A. Bekrar, T. Berger, and D. Trentesaux, "The control of myopic behavior in semi-heterarchical production systems: A holonic framework," *Eng. Appl. Artif. Intell.*, vol. 26, no. 2, pp. 800–817, Feb. 2013.
- [14] M. Rolón and E. Martínez, "Agent-based modeling and simulation of an autonomic manufacturing execution system," *Comput. Ind.*, vol. 63, no. 1, pp. 53–78, Jan. 2012.
- [15] F. Maturana, W. Shen, and D. H. Norrie, "MetaMorph: An adaptive agent-based architecture for intelligent manufacturing," *Int. J. Prod. Res.*, vol. 37, no. 10, pp. 2159–2173, Jul. 1999.
- [16] S. S. Heragu, R. J. Graves, Byung-In Kim, and A. St Onge, "Intelligent agent based framework for manufacturing systems control," *IEEE Trans. Syst. Man Cybern. - Part Syst. Hum.*, vol. 32, no. 5, pp. 560–573, Sep. 2002.
- [17] C. Ou-Yang and J. S. Lin, "The development of a hybrid hierarchical/heterarchical shop floor control system applying bidding method in job dispatching," *Robot. Comput.-Integr. Manuf.*, vol. 14, no. 3, pp. 199–217, 1998.
- [18] J. S. Cox and E. H. Durfee, "Discovering and Exploiting Synergy Between Hierarchical Planning Agents," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, New York, NY, USA, 2003, pp. 281–288.
- [19] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference architecture for holonic manufacturing systems: PROSA," *Comput. Ind.*, vol. 37, no. 3, pp. 255–274, 1998.