

Dynamic Service Reconfiguration with Multi-agent Systems

Nelson Rodrigues, Paulo Leitão and Eugénio Oliveira

Abstract Most modern manufacturing systems rely on constantly seeking new solutions to better fulfil their manufacturing objectives. As reported in today's manufacturing literature, dynamic service reconfiguration is one solution that permits to endorse continuous service reconfiguration, flexibility and evolvable systems. In spite of the current research efforts, real reconfiguration solutions are still lacking automated tools that support dynamic and runtime reconfigurations by discovering new adaptation needs and opportunities and, thus, explore possible actions leading to new system configurations. To overcome these issues, it is essential to provide solutions that answer to the “when” and “what” to reconfigure questions. Most of the service changes triggers rely on reactive events, where decisions come from a centralized decision-maker and are performed manually. Based on these facts a service-oriented multi-agent systems architecture is described aiming at actively promoting service reconfiguration (e.g., improvement of the service's properties and/or update the services' catalogue) to cope with the unexpected and unpredictable condition changes. This paper describes the processes that decide which service reconfiguration should be applied to each circumstance. The developed prototype for a flexible manufacturing system case study allowed verifying the feasibility of the proposed dynamic service reconfiguration solution in different scenarios.

N. Rodrigues (✉) · P. Leitão · E. Oliveira
LIACC—Artificial Intelligence and Computer Science Laboratory,
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
e-mail: nrodrigues@ipb.pt

P. Leitão
e-mail: pleitao@ipb.pt

E. Oliveira
e-mail: eco@fe.up.pt

N. Rodrigues · P. Leitão
Polytechnic Institute of Bragança, Campus Sta Apolónia, 5300-253 Bragança,
Portugal

N. Rodrigues · E. Oliveira
Faculty of Engineering—University of Porto, Rua Dr. Roberto Frias,
4200-465 Porto, Portugal

Keywords Service reconfiguration • MAS • Flexible manufacturing system

1 Introduction

The growing of global markets and interest of customer satisfaction challenge manufacturing companies to deliver high-quality customized products while facing the growing requirements from the customers. To face this problem, several governmental initiatives are promoting the research and development towards the factories of the future, e.g., Industrie 4.0 [1] promoted by the German government that is recognized as the 4th industrial revolution. In these visions, digitization of manufacturing and particularly the Internet of Things and Services are recognized as crucial to support the deployment of more flexible, robust, responsive and reconfigurable systems.

Several reconfigurable paradigms were proposed during the last decades, introducing flexible and agile characteristics to react promptly to unexpected events, system failures, quality deviations, etc., avoiding the loss of financial revenue and trust from the customer viewpoint [2]. As an example, reconfigurable manufacturing system (RMS) [3] is a well know paradigm that provides the ability to repeatedly change and reorganize the components of a system in a cost-effective way. Consequently, the reconfiguration of a system is embraced when it brings benefits to the manufacturing control context.

More particularly in service-oriented manufacturing approaches the functionalities of manufacturing components are offered and consumed as services [2]. For example, ANEMONA-S+Thomas proposes a framework to implementing a service-oriented intelligent manufacturing system [3]. The current works of service-oriented manufacturing systems constitute the fundamental functionalities to cope the service reconfiguration problem. The adaptation of the system behaviour and its evolution are based in the service readjustment of the manufacture elements or settings of a system. Each one of the service reconfiguration is composed of several services; an automated service composition is proposed in [4], with the objective to maximize the overall quality of the final compositions using agents that adapt services processes, in a continuous form. A dynamic service reconfiguration is proposed by [5] that explores the use of agents to achieve consistent service reconfiguration solutions focusing fault-tolerant systems.

In addition to the software reconfiguration, hardware reconfiguration plays an important role. An approach that considers software and hardware reconfiguration is proposed by [6], employing a knowledge ontology and AI-planning for the service reconfiguration. In the manufacturing context, some projects already addressed the service reconfiguration problem, namely SOCRADES [7] that is oriented to the reconfiguration of smart embedded devices and PRIME [8] that lies on a plug and produce system architecture capable of reacting to unexpected disturbances to maintain the productivity and quality parameters. The IDEAS project [9] supports the use of reconfigurable production systems using agent technology to

perform the on-line reconfiguration without the need of reprogramming efforts, and more recently, the PERFoRM project [10] is developing a system architecture for the seamless reconfiguration of machinery and robots as response to operational or business events.

However, most of the real service reconfiguration solutions are executed manually, reactively and in a centralized perspective. In fact, one drawback is that the decisive actions for the system reconfiguration are made after the occurrence of failure and sometimes involves stopping a running system, reconfiguring and then restarting the system which does not go according to industrial needs, pointing out to have a pro-active, distributed and online service reconfiguration.

Having this in mind, the paper describes a service reconfiguration approach that allows the identification of the opportunities for reconfiguration, in a pro-active and online manner, determines and implements on-the-fly the best strategies for the service reconfiguration that will lead to better production efficiency. For this purpose, a multi-agent system (MAS) is providing distributed intelligence to run the service reconfiguration, particularly embedding intelligent mechanisms for the early detection of reconfiguration opportunities, e.g. anticipating performance or quality degradation, and also advanced data analytics to support the service reconfiguration by selecting strategies for improving the service properties (e.g., QoS and execution time) or updating the catalogue of offered services (e.g., offering a new service that has more demand).

The remaining of this paper is organized as follows. Section 2 overviews the multi-agent system architecture for dynamic service reconfiguration, particularly addressing the “when and how to reconfigure” phases. Section 3 describes the mechanism for determining the alternative solutions for service reconfiguration and Sect. 4 presents the evaluation methodology to select the best service reconfiguration solution from the space of solutions previously created. Section 5 presents some preliminary experimental results aiming at validating the proposed approach. Finally, Sect. 6 rounds up the paper with the conclusions.

2 Multi-agent System Architecture for Service Reconfiguration

2.1 Agents to Provide Intelligence Supporting Service Reconfiguration

The literature suggests Service-Oriented Architecture (SOA) [2] as an excellent solution to face the many current industry challenges, namely providing interoperability in heterogeneous systems. The development of SOA-based solutions requires the implementation of several features, namely service-discovery, service-registration, service-composition and service-reconfiguration. In particular, service reconfiguration is crucial to facilitate the changes taking place

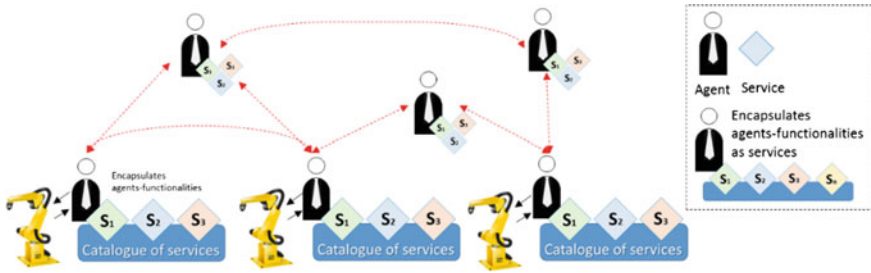


Fig. 1 Service-oriented multi-agent system

in unpredictable environments. As such, this task requires the development of an adaptive system that can regulate service management in response to events and changes in the environment, and particularly a continuous monitoring to dynamically support any change that might occur by reacting on-the-fly. Having this in mind, a MAS is proposed to embed intelligence and adaptive mechanisms in the distributed and autonomous agents to support the service reconfiguration on-the-fly, as shown in Fig. 1.

The service-oriented MAS architecture considers a network of intelligent and autonomous agents, each one exposing its functionality as services. These services are published in a repository and can be reached via discovery mechanisms. The question that arises is that, over the time, services can become less competitive (i.e. not being requested), e.g., due to their low QoS or high price, requiring the execution of proper actions to improve the services competitiveness. For this purpose, a continuous monitoring of the service performance is required to identify when a reconfiguration should be performed, and intelligent mechanisms should be implemented to determine the best solution for the service reconfiguration. In this approach, the service reconfiguration procedure is performed in a distributed way, with individual agents embedding intelligent mechanisms to run the reconfiguration of their catalogue of services.

2.2 Service Reconfiguration Approach

According to the architectural principles, each individual agent is running an online mechanism aiming at continuously collecting data, identifying opportunities to reconfigure, determining how the reconfiguration can be performed and deciding whether the reconfiguration should be performed, as illustrated in Fig. 2.

In this service reconfiguration approach, the first phase is related to the continuous collection of up-to-date information on the services' performance and the storage of the relevant data into a local database. A service reconfiguration trigger relies on the determination of when is the best moment to reconfigure, which can happen in different moments of the production cycle, e.g., in a case of performance or quality degradation, failure occurrence on the introduction of new products.

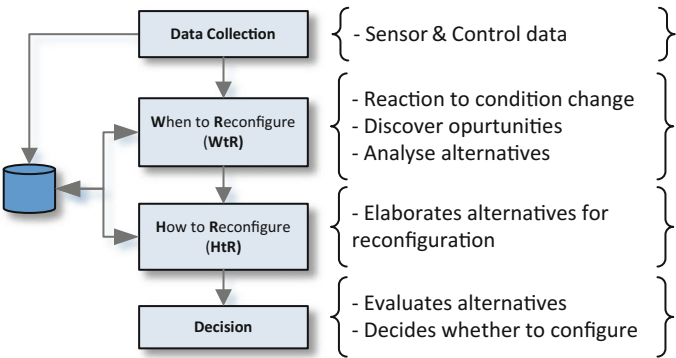


Fig. 2 Service reconfiguration module implemented in each agent

Table 1 Possible types of service reconfiguration

Reconfiguration types	Description	Effort
Improve the service’s behaviour	Improve the behaviour of the service without replacement, e.g., calibrating tools and switching components of the process that executes the service aiming at reducing service time or improve the service quality	Low
Change the service’s catalogue	The catalogue of offered services is changed, i.e. new services are provided by the existing agent or by new agents, e.g., offering a new drilling service	High

For this purpose, the *When to Reconfigure* (WtR) module is responsible for monitoring and analysing the collected data in order to identify the triggers for the reconfiguration. In the proposed approach, this model relies on events, periodic and trend triggering strategies to support the on-the-fly reconfiguration (see [11] for more details about the WtR module).

After being identified an opportunity to reconfigure, the *How to Reconfigure* (HtR) module determines how the service reconfiguration can be implemented. The process comprises the building of a pool of possible alternatives for the service reconfiguration, followed by a semantic checking that reduces the dimension of the alternative solutions. The elaboration of the alternative solutions considers two classes of service reconfiguration, namely improving the service’s behaviour (as a weak-reconfiguration class) and changing the service’s catalogue (strong-reconfiguration class), as shown in Table 1.

The decision module is responsible for evaluating the effectiveness of the service reconfiguration alternatives, considering different criteria set by the system managers. Only after recognizing the expected profits of such service reconfiguration, the selected solution is implemented. In this work, special attention is devoted to the HtR module.

3 Mechanism to Create Alternative Solutions

The HtR module contains a mechanism to create alternative solutions that works in two phases. The first phase is responsible for the creation of a pool of alternative service reconfiguration solutions and the second phase is in charge of testing the compatibility of the alternative solutions by using the semantic matching.

3.1 Build a Space of Alternative Solutions Phase

In order to produce several reconfiguration alternatives, each individual agent, after receiving the reconfiguration triggers from the WtR module can recommend the improvement/replacement of a specific service, as shown in Fig. 3, considering the pool of available services not installed (i.e. they exist, but are not offered at the moment).

The algorithm embedded in each agent to calculate the alternative solutions for the service reconfiguration is represented as follows.

```

 $S_1 \leftarrow$  select the worst service from the catalog
If ((utilization rate of Service  $S_1 \geq \alpha_1$ ) OR (rate of service bids of  $S_1 \geq \alpha_2$ ))
  Improve the Service( $S_1$ )
Else
   $S_{new} \leftarrow$  select the potential best Service
  If (rate of service bids of  $S_{new} \geq$  rate of service bids of  $S_1$ )
    Perform Service Replacement (remove  $S_1$ , add  $S_{new}$ )
  
```

A service with weak performance can be improved if its utilization rate or the missing bids for its usage are higher than certain threshold values (α_1). This can involve the execution of a set of actions regarding the optimization of the process

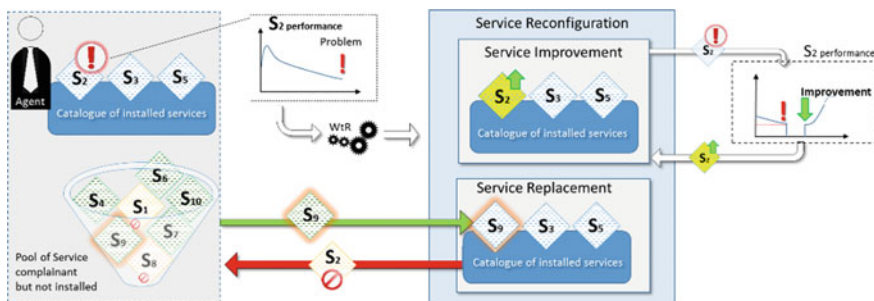


Fig. 3 Service reconfiguration alternatives (service replacement and service improvement)

encapsulated by the service, e.g., calibrating tools, optimizing operational parameters or replacing components. Otherwise, the best services from the pool of available services and not installed are selected to create alternative possibilities to replace the one with the weak performance.

Additionally, sometimes it is useful to consider services that are not available in order to discover new opportunities for reconfiguration. The decision to explore potential solutions is calculated by the nervousness control that adjusts the threshold values (i.e. α_1 , α_2). Moreover, the learning module is capable of changing the exploration rate value, allowing controlling the exploration of different solutions. This process can generate an enormous volume of service configuration alternatives, resulting in a time-consuming process. In this way, the agent can run this process in the background, especially when the trigger for reconfiguration follows the periodic strategy.

3.2 Semantic Matching Phase

Inspired on the service description topology (i.e., manufacturing-service model [12]) the technical operator describes semantically each service, resource, and process that exists on the system (e.g., describing in a RDF/XML format). Therefore, each agent contains the entire device’s information in its catalogue of services (e.g., gripper’s characteristics).

Moreover, it also contains information about which processes the agent can produce (e.g., the resource r can make process *openGripper* using the service *gripper1*) under some constraints (e.g., physical limitations and QoS).

Figure 4 illustrates the particular agents, representing industrial robots, which are implementing the semantic reasoning about the logical configuration, for



Fig. 4 Semantic matching of service reconfiguration solutions

example, using JENA, to determine the feasibility of the service reconfiguration solution (i.e. semantic matching between the resource machines and all services from the pool of services).

The outcome of this process is set of feasible service reconfiguration solutions.

4 Evaluation of Service Reconfiguration Alternatives

The process of selecting the optimal system configurations is not consensual in literature. One novelty, besides the generation of alternative service reconfiguration solutions, is the distributed measurement of the reconfiguration alternatives effectiveness.

4.1 Evaluation of the Service Reconfiguration Phase

In general, an evaluation takes into account several criteria, e.g., processing time and quality, which can be evaluated individually or combined. The agents conduct the evaluation of the service reconfiguration solutions based on two metrics:

- Maximizing the service composition quality, which permits to select the service with the highest quality (see Table 2).
- Minimizing the reconfiguration index which permits to select the service configuration with best improvement values with minimal implementation effort.

The quality metrics denoted in Table 2 were based on the MAS architecture capable of evaluating, in run-time, several hypotheses of services composition [13].

In a next step, ranking criteria are created based on the selected indicators for the created configurations. Note that the measurements of these indicators can change over time according to each system manager’s requirements.

In contrast, it is essential to understand the reconfiguration effort, i.e., to calculate for each potential solution the reconfiguration cost. However, these

Table 2 Representation of the agent and service variables—from the machine viewpoint

Name and equation	Description
QoS availability $f(\phi) = \frac{\sum \lambda}{\sum \lambda + \sum \psi} * 100$	The ratio of the service uptime of time period, where λ stands for service uptime and ψ for the service downtime
QoS response time $f(\theta) = \delta - \rho$	Performance of a service. Given by the difference between conclusion time δ , and ρ the request time
QoS throughput $f(\eta) = \frac{\sum \gamma}{\sum t}$	Provider performance index. Given by the maximum number of services to process a unit of time where γ stands for the complete request and t for the unit time

requirements do not support the dimension of the reconfiguration effort index. In the literature, the structure matrix analysis investigates the capability to reconfigure [14]; however the process relies on centralized decisions. This indicates a lack of research in measuring the reconfigurability effort and its impact on the decentralized way. Thus, the proposed model inspired from [15] and [13], takes a step forward by joining the following indicators.

The reconfiguration index (RI) considers the number of reconfigurations. Each agent contains a vector with the actual configuration of the services (CC) that are being currently executed. In addition, the agent contains other simulated vectors that represent the alternative configurations previously built (AR). The idea is to compare the CC vector and the simulated AR vector to understand the effort that is required [15].

$$RI = 1 - \frac{\sum_{s_i} (\text{modificationCost}(s_i))}{\# \text{ of services}} \quad (1)$$

where

$$\text{modificationCost}(s_i) = \begin{cases} 1, & \text{if } (cc[s_i] = AR[s_i]) \\ 0, & \text{otherwise} \end{cases}$$

According to Formula (1), an evaluation process is created by the number of modifications. For example, if the alternative is equal to the current configuration, the RI is 0; RI becomes close to 1 as many modifications exist. Besides the RI, the reconfiguration cost is another important measure, as the Formula (1) considers the same weight for different modifications. The reconfiguration cost (RC) [16] considers different service modification costs between the services, calculated as follows:

$$RC = nr \times \sum_{s_i} (\text{modificationCost}(s_i) + \text{lbcost}(s_i)) \quad (2)$$

where nr represents the number of modification multiplied by the unitary cost of modifying the service s_i , which includes the $\text{modificationCost}(s_i)$ of a particular service s_i and labor cost $\text{lbcost}(s_i)$. The reconfiguration effort is evaluated with this simple metric, which considers the number of modifications required for a specific service reconfiguration. The positive impact is defined if the expected profit is higher, which is obtained as shown in Formula (3), where the *Expected Benefit* value is being calculated at the planning phase:

$$\text{Expected Profit} = \text{Expected Benefit} - RC \quad (3)$$

At the end the list of solutions is ordered according to an assessment with low reconfiguration costs, high levels of benefits ensuring high quality.

4.2 Implementation Phase

After the evaluation process, the *Decision* module is responsible for deciding if the best service reconfiguration alternative is implemented or not. Such decision takes into account several points, namely the expected benefit and quality of such service reconfiguration. Thus the automatic mechanism considers self-* properties like self-learning to support the decision-making about alternative service reconfiguration scenarios; in parallel this module provides valuable information in proper graphical user interfaces.

5 Experimental Results

The proposed approach for service reconfiguration was tested using a flexible manufacturing system case study comprising a set of 6 workstations (WS), interconnected through conveyors, each WS offering a limited set of operations (i.e. services). Several sub-products are made in this system, namely the parts in the form of the letters A, B, E, I, L, P and T, which combined can produce the final products BELT and AIP. More detail about the case study benchmark can be found in [17].

For this purpose, the proposed MAS was implemented using the JADE framework. Several agents were launched to represent several WS and the products requested to be manufactured in the system. Each one of the WS agents has embedded the WtR module (to identify opportunities to reconfigure) and the HtR (to determine the best strategy to reconfigure). In the proposed case study, the service reconfiguration in WS-3 is considered when the service's quality is not meet (trend) and also due to the occurrence execution failures (event). Each WS agent has a catalogue of 3 installed services and 2 services available but not installed. The time to replace one service installed by another service available is 60 s, the time to improve the service performance is 30 s. and the service recovery 120 s.

Table 3 summarizes the results considering the execution of different scenarios, each one considering different batch sizes.

Table 3 Experimental results

Scenarios	Cmax in s (without reconfiguration)	Cmax in s (with reconfiguration)	Improvement (%)	Details
#5 (BELT)	1595	1565	1.8	#1 change, #0 replace
#10 (BELT)	3234	3147	2.6	#1 change, #0 replace
#15 (BELT)	5245	5068	3.3	#2 change, #1 replace
#20 (BELT)	7576	7268	4.0	#1 change, #2 replace

The comparative analysis of each scenario is performed using the value of the makespan, known as C_{max} . Once the reconfiguration decision is made according to the expected profit as is formulated in Formula (3), the value C_{max} allows comparing different scenarios without performing reconfiguration, illustrating the number of performed changes and replacements.

The analysis of the experimental results shows the benefits of considering the proposed approach. In fact, all scenarios reported improvements in the C_{max} values ranging from 1.8–4.0% when considering the automatic service reconfiguration. It is also worth noting that the bigger is the batch size the better is the achieved improvement.

6 Conclusions and Further Work

Companies are successfully using centralized techniques for analyzing up-to-date information collected from the shop floor to identify service failures or performance deviations. Once the need for service reconfiguration recognized, its implementation is often carried out in a manual mode, usually held as a recovery approach. With the objective to increase the opportunities to dynamically evolve and implement the service reconfiguration on-the-fly, this paper proposes an agent-based approach for the dynamic, distributed efficient and on-the-fly service reconfiguration. The proposed decentralized approach pro-actively generates several service reconfiguration solutions promoted by different triggering strategies. The developed modules are embedded in smart agents, that do not only recognize opportunities to change, but also assist engineers in exploring and deciding about different alternative configuration possibilities, to cope with disturbances or predicting production changeover. Another contribution of this work is the evaluation of the potential service reconfiguration solutions. The preliminary experimental results validate the feasibility of the mechanism that determines how to reconfigure services offered by the system leading to more efficient and agile systems.

As future work, there are some open questions that require a deep analysis, e.g., what would happen if all agents representing manufacturing resources adapt at the same time, and what are the necessary rules to control the system nervousness avoiding falling into a chaotic system.

References

1. Kagermann, H., Wolfgang, W., Helbig, J.: Securing the future of German manufacturing industry. Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report of the Industrie 4.0 Working Group (2013)
2. MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P.F., Metz, R.: Reference model for service oriented architecture 1.0. OASIS Stand. **12**, 1–28 (2006)

3. Giret, A., Botti, V.: ANEMONA-S+Thomas: a framework for developing service-oriented intelligent manufacturing systems. In: *Service Orientation in Holonic and Multi-agent Manufacturing*, pp. 61–69. Springer International Publishing (2015)
4. Lécué, F., Gorrongoitia, Y., Gonzalez, R., Radzinski, M., Villa, M.: SOA4All: An innovative integrated approach to services composition. *2010 IEEE International Conference on Web Services (ICWS)*, pp. 58–67 (2010)
5. Tsai, W.T., Song, W.W., Chen, Y.N., Paul, R.: Dynamic system reconfiguration via service composition for dependable computing. *Reliable Syst. Unreliable Network. Platf.* **4322**, 203–224 (2007)
6. Yang, T.-H., Lee, W.-P.: Intelligent service reconfiguration for home robots. In: Ding, X., Kong, X., Dai, S.J. (eds.) *Advances in Reconfigurable Mechanisms and Robots II*, pp. 735–745. Springer International Publishing, Cham (2016)
7. Colombo, A.-W., Karnouskos, S., Mendes, J.-M.: Factory of the future: a service-oriented system of modular, dynamic reconfigurable and collaborative systems. In: Benyoucef, L., Grabot, B. (eds.) *Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management*. Springer, London, pp. 459–481 (2010)
8. Rocha, A., Di Orio, G., Barata, J., Antzoulatos, N., Castro, E., Ratchev, S.S., Ribeiro, L.: An agent based framework to support plug and produce. In: *12th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 504–510 (2014)
9. Onori, M., Lohse, N., Barata, J., Hanisch, C.: The IDEAS project: plug & produce at shop-floor level. *Assem. Autom.* **32**, 124–134 (2012)
10. Leitão, P., Barbosa, J., Pereira, A., Barata, J., Colombo, A.: Specification of the PERFORM architecture for the seamless production system Reconfiguration. In: *42nd Annual Conference of the IEEE Industrial Electronics Society (IECON)* (2016)
11. Rodrigues, N., Leitão, P., Oliveira, E.: Triggering strategies for automatic and online service reconfiguration. In: *11th Conference*, pp. 76–82 (2016)
12. Gamboa Quintanilla, F., Cardin, O., L’Anton, A., Castagna, P.: A modeling framework for manufacturing services in service-oriented holonic manufacturing systems. *Eng. Appl. Artif. Intell.* **55**, 26–36 (2016)
13. Rodrigues, N., Leitão, P., Oliveira, E.: Self-interested service-oriented agents based on trust and QoS for dynamic reconfiguration. In: *Service Orientation in Holonic and Multi-Agent Manufacturing (SOHOMA’14)*, Nancy, France (2014)
14. Farid, A.M.: Measures of reconfigurability and its key characteristics in intelligent manufacturing systems. *J. Intell. Manuf.* 1–17 (2014)
15. Neves, P.: Reconfiguration methodology to improve the agility and sustainability of plug and produce systems (2016)
16. Zhou, J., Djurdjanovic, D., Ivy, J., Ni, J.: Integrated reconfiguration and agent-based preventive maintenance decision making. *IIE Trans.* **39**(12), 1085–1102 (2007)
17. Trentesaux, D., Pach, C., Bekrar, A., Sallez, Y., Berger, T., Bonte, T., Leitão, P., Barbosa, J.: Benchmarking flexible job-shop scheduling and control systems. *Control Eng. Pract.* **21**(9), 1204–1225 (2013)