Computers in Industry 101 (2018) 81–90

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Decentralized and on-the-fly agent-based service reconfiguration in manufacturing systems

Nelson Rodrigues[a,b,c,*], Eugénio Oliveira[b,c], Paulo Leitão[a]

[a] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Braganca, Portugal
[b] Faculty of Engineering - University of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
[c] LIACC - Artificial Intelligence and Computer Science Laboratory, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

ARTICLE INFO

Keywords:
Service reconfiguration
Multi-agent systems
Service-oriented architectures
Intelligent manufacturing systems

ABSTRACT

Intelligent manufacturing systems rely on the capability to adapt and evolve to face the volatility of dynamic markets. The complexity of these systems increases with the demand of more customized and quality products, which requires more agile and flexible methods to support the dynamic and on-the-fly system reconfiguration aiming to respond quickly to product changes, by offering more efficient services. In this service-oriented manufacturing context, where process functionalities are modelled as services (e.g., quality control, welding and transportation), the dynamic reconfiguration of the services structure (e.g., in terms of quality, processing time and provided features) assumes a critical role to achieve the referred requirements. Despite the current research efforts, the service reconfiguration approaches usually use reactive event triggers, with decisions coming from a centralized decision-maker and performed manually. This means a lack of dynamic and run-time reconfiguration flexibility by discovering opportunities and needs to change, and, thus, exploring possible actions leading to new and appropriate system configurations. To overcome the mentioned issues, it is essential to provide solutions that answer to when and how to reconfigure a manufacturing system in an integrated, automatic and dynamic manner. For this purpose, this paper introduces an agent-based approach for service reconfiguration in manufacturing systems that allows the identification of opportunities in a pro-active and dynamic manner, and the on-the-fly implementation of new configuration solutions leading to a better production efficiency. The experimental results, using a flexible manufacturing system case study, allowed to verify the feasibility and benefits of the proposed agent-based service reconfiguration solution for competitive and collaborative industrial automation scenarios.

1. Introduction

Manufacturing industry is facing strong demands, in terms of products quality, customization and delivery time, imposed by the global market growth, which requires the capability to react rapidly and cost-effectively to condition changes. To address the mentioned issues and aligned with the vision of Industry 4.0 [1], Internet of Things and Internet of Services, established upon Cyber-Physical Systems (CPS) and complemented with other emergent ICT technologies, such as Big data, cloud computing and data analytics, are recognized as crucial to support the deployment of more flexible, robust, responsive, reconfigurable and interoperable systems. In this context, the development of Service-Oriented Architectures (SOA) [2] based solutions, requires the implementation of several features, namely service-discovery, -registration, -composition and -reconfiguration. In particular, the service reconfiguration is crucial to respond to condition changes that take place in unpredictable environments by dynamically adapt and improve the functionalities abstracted by the offered services. Usually, the service reconfiguration is performed to cope with unexpected condition changes, to improve the system competitiveness and to respond to new business strategies. Note that in manufacturing systems, services can be seen as functionalities offered by a device or system, e.g., pick-and-place or welding operations provided by a robot or an inspection operation provided by a quality control station.

Despite the significant research efforts found in literature, the existing service reconfiguration approaches are focusing limited scopes, e.g., components' replacement (to react to the harmful effects or breakdowns) and re-planning (to deal with the modified configurations' requirements) [3], and lack the automatic, dynamic and run-time reconfiguration procedures that enable discovering new reconfiguration

* Corresponding author at: Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Braganca, Portugal.
E-mail addresses: nrodrigues@ipb.pt (N. Rodrigues), eco@fe.up.pt (E. Oliveira), pleitao@ipb.pt (P. Leitão).

https://doi.org/10.1016/j.compind.2018.06.003
Received 31 October 2017; Received in revised form 15 May 2018; Accepted 23 June 2018
0166-3615/

opportunities and exploring new system configurations. Additionally, the deployed service reconfiguration approaches are usually performed manually and reactively in a centralized manner, usually requiring to stop a running process, diagnose the problem, perform the re-configuring and restart the system/device. However, the determination of all possible service configurations' solutions in a repeated and consistent manner goes beyond the human capability in an acceptable time. In this sense, reconfiguring the service manually and then restarting the system is not acceptable if we want to comply with the dynamics and requirements of current industrial needs [4]. As a consequence, new service reconfiguration approaches are required to support the truly reconfiguration procedure by being performed automatically, dynamically and on-the-fly. Note that in industrial manufacturing systems, reconfiguration procedures are strongly dependent on physical resource constraints and response time.

Having this in mind, the paper describes a service reconfiguration approach for manufacturing systems that allows the pro-active and dynamic identification of opportunities for reconfiguration and the on-the-fly implementation of the best strategies for the service re-configuration that will lead to an increase of the production efficiency. For this purpose, multi-agent system (MAS) is used to distribute the system intelligence aiming to run the service re-configuration process in an autonomous, modular and decentralized manner. The proposed MAS approach is enriched with intelligent mechanisms for the early detection of reconfiguration opportunities, e.g., a performance degradation or the introduction of a new product, and the selection of re-configuration strategies for improving the service quality or updating the catalogue of offered services. The conflicts arising from the dynamic reconfiguration provided by the distributed agents, acting in a selfish mode, are avoided by considering a collaborative mechanism that considers the interests of the individual agents together with the interests of the overall system. This approach was tested in an experimental flexible manufacturing system, and the results show the benefits of this dynamic and pro-active service reconfiguration approach.

The remaining of this paper is organized as follows. Section 2 overviews the service reconfiguration concept and the existing related work and establishes the requirements for a truly dynamic, intelligent and pro-active service reconfiguration process. Section 3 introduces the MAS architecture for the dynamic service reconfiguration approach, particularly describing the "when" and "how" reconfiguration phases. Section 4 describes the process to select the best service reconfiguration alternative from the built space of solutions and introduces a mechanism to evaluate the pertinence of these solutions from a collaborative perspective. Section 5 describes the experimental case study and the implementation details, and analyses the experimental results. Finally, Section 6 rounds up the paper with the conclusions.

2. Related work

The SOA paradigm [5] is based on the concept of software systems offering and consuming services, each one encapsulating the functionalities of a service provider. The use of service-orientation brings significant benefits to design complex systems allowing to face interoperability and loose-coupled abstraction, being currently, amongst other areas, applied to model and design modern manufacturing systems [34], where services can encapsulate physical operations like welding and pick-an-place. In these systems, the concepts of services aggregation, composition, and orchestration are important, and strongly impact the service reconfiguration capability. Note that services can be composed of atomic services (i.e. simple and indivisible services) following a specific workflow.

Basically, service reconfiguration is related to the adaptation of the existing services to deal with unexpected internal or external condition changes, such as failure of a service and loss of the quality of service (QoS) [3]. Although focusing primarily on fault mitigation, service re-configuration may also consider reconfiguration opportunities that are not initially described but contribute to improve the system performance. In fact, over the time, services can become less competitive, e.g., not being requested by service consumers, requiring the execution of proper actions to improve their competitiveness. To exploit that situation, several types of services reconfiguration can be identified:

- **Improvement of the service's behaviour:** the functionality of the service is improved aiming to increase its performance, e.g., calibrating or switching components; this can be seen as a weak service reconfiguration and only needs an internal perception of the problem.
- **Changing the services' catalogue:** the catalogue of services offered by an entity can be modified to face the service demand through adding new services and/or removing others, e.g., a robot that changes a tool to be able to weld metal sheets; this can be seen as a strong service reconfiguration and needs to have an external perception of the problem.
- **Changing the structure of a composed service:** a composed service can be re-organized by changing the structure and workflow of atomic services (known as service choreography), e.g., adding or removing services to better accommodate the evolution in the available atomic services; this can be seen as a strong service re-configuration and also needs to have an external perception of the problem.

The use of dynamic service orchestrators can provide these features, particularly to dynamically create workflows when required. Complementary, aiming to execute a dynamic, pro-active and on-the-fly service reconfiguration, considering the referred reconfiguration types, the following requirements need to be observed [6]:

- R1: The opportunity to execute a service reconfiguration must be identified internally (regarding the system), automatically and at run-time.
- R2: The system needs to have the capability to select an alternative reconfiguration solution and perform the reconfiguration on-the-fly, reducing the impact of condition changes (e.g., a performance degradation).
- R3: The service reconfiguration must be performed according to the "nervousness" state of the system, i.e. reacting smoothly or dramatically in such a way that mitigates the problem.
- R4: The service reconfiguration process should be performed in a distributed manner and consider competitive and collaborative scenarios.

In the literature, a significant part of the research in service re-configuration is devoted to the change in the service composition aiming to compose the best service that meets the client's requirements. This composition can be performed in two distinct moments: design-time and run-time. Most of the works consider that reconfiguration is planned in the design phase, setting up the system to cope with expected changes [7–9]. In contrast, run-time reconfiguration reacts promptly to situations during the system execution to overcome unexpected events that were not considered at design-time, as illustrated by [10–12]. However, the reconfiguration is usually based only on the migration of services that are considered as a reconfiguration action in an automatic manner. The selection of different composition instants, in essence, remains a challenge creating a design-time/run-time trade-off situation [13]. Particular scenarios result in complex computational problems that cannot be efficiently solved in run-time and, in such cases, a design-time policy might be more beneficial in the long run, where time is not a constraint to apply more complex algorithms.