

Towards Self-organized Service-Oriented Multi-agent Systems

Paulo Leitão

Polytechnic Institute of Bragança, Campus Sta Apolónia, Apartado 1134,
5301-857 Bragança, Portugal
pleitao@ipb.pt

Abstract. The demand for large-scale systems running in complex and even chaotic environments requires the consideration of new paradigms and technologies that provide flexibility, robustness, agility and responsiveness. Multi-agents systems is pointed out as a suitable approach to address this challenge by offering an alternative way to design control systems, based on the decentralization of control functions over distributed autonomous and cooperative entities. However, in spite of their enormous potential, they usually lack some aspects related to interoperability, optimization in decentralized structures and truly self-adaptation. This paper discusses a new perspective to engineer adaptive complex systems considering a 3-layer framework integrating several complementary paradigms and technologies. In a first step, it suggests the integration of multi-agent systems with service-oriented architectures to overcome the limitations of interoperability and smooth migration, followed by the use of technology enablers, such as cloud computing and wireless sensor networks, to provide a ubiquitous and reconfigurable environment. Finally, the resulted service-oriented multi-agent system should be enhanced with biologically inspired techniques, namely self-organization, to reach a truly robust, agile and adaptive system.

Keywords: Multi-agent systems, Service-oriented systems, Self-organization, Cyber-physical systems.

1 Introduction

The global markets are imposing strong changing conditions for companies running their businesses, sometimes comprising complex and large scale systems. The way such systems operate is crucial for the companies' competitiveness and sustainability. As examples, the electrical smart grids advent are imposing a set of new requirements and opportunities to the way of managing distributed energy systems, and the lack of efficient and adaptive traffic control systems provokes billions of lost hours and billions of litres of wasted petrol cost having a huge impact on the air quality [1]. Also in the manufacturing sector, which is not as powerful as it once was but is still a driving force in economy, the higher variety of products, the possible large fluctuations in the

demand, the shorter lifecycle of products expressed by a higher dynamics of new products, are some challenges that manufacturing companies have to deal with to remain competitive.

The challenge is to consider paradigms and technologies to allow the development of such distributed, large-scale complex systems that behave in a better manner. Of special interest in this work are the cyber-physical systems (CPS), which are a strategy to address the challenge of developing solutions combining mechatronics and ICT technologies, to face the current requirements imposed by customers acting in a global economy [2]. The use of CPS aims at increasing implementation of large-scale systems, improving the adaptability, autonomy, efficiency, functionality, reliability, safety, and usability of such systems. The development of CPS can be performed by using several promising technologies, namely multi-agent systems (MAS) [3], service-oriented architectures (SOA) [4] and IEC 61499 [5]. Several applications of these technologies (mainly prototypes and laboratorial ones) are reported in the literature (see [6-8] as examples for agent-based solutions). However, the generalized idea is that the individual application of these paradigms does not completely solve the current problems imposed by industry since each one introduces important and valuable features, but miss other important ones.

A new integrated vision combining complementary emergent technologies allows reaching control structures with distributed intelligence supporting the enterprise integration (vertical and horizontal dimensions) and running in truly distributed and ubiquitous environments. Additionally, the enrichment of these distributed systems with mechanisms inspired by biology supports the dynamic structure re-configuration, thus handling more effectively with condition changes and unexpected disturbances, and minimizing their effects. As an example, the integration of service-oriented principles with MAS allows to combine the best of the two worlds, and in this way to overcome some limitations associated to multi-agent systems, such as interoperability.

This integrated vision can be a valuable help in the demand of important questions and problems that raise when running this kind of systems, namely how the global optimization is achieved in decentralized systems, how temporary hierarchies are dynamically formed, evolved and removed, how individual entities self-organize and evolve to support evolution and emergency, and how to adapt their emergent behaviour using learning algorithms.

Having this in mind, this chapter discusses and explores a new, integrated perspective for the engineering of complex and adaptive systems, as cyber-physical systems are. This new perspective is based on four main steps: i) distribute the complex problem in several individual, simple entities, ii) connect the distributed entities to support the emergence of a global system, iii) provide evolvable mechanisms to support dynamic adaptation, and finally iv) provide technology enablers to support the operation of these systems in a ubiquitous environment. For this purpose, and after combining multi-agent systems with new emergent complementary paradigms and technologies, biologically inspired techniques should be considered regarding the truly self-adaptation and evolution of complex systems. Particularly the use of self-organization mechanisms, combined with the service-oriented multi-agent systems, allows designing innovative, flexible, adaptive, responsive and reconfigurable production control

systems that better address the current industrial requirements. A particular attention will be devoted to discuss how these self-organized, self-adaptive and reconfigurable systems can be engineered and deployed.

The rest of the paper is organized as follows: Section 2 discusses a new vision for the engineering of adaptive cyber-physical systems based on a 3-layer framework. Section 3 presents the basis for the engineering of service-oriented multi-agent systems and Section 4 discusses the enhancement of these systems by applying technology enablers to achieve ubiquity and self-* properties to achieve adaptation and evolution. Section 5 discusses the actions to be implemented for a wider adoption of such distributed and adaptive control structures by industrial players. Finally, Section 6 rounds up the paper with the conclusions.

2 Engineering Adaptive, Complex Cyber-Physical Systems

The demand for intelligent, flexible and adaptive control systems that exhibit high degree of re-configurability imposes strong requirements on the way the systems are designed, installed and operated. In spite of the promising perspective of some emergent paradigms and technologies such as multi-agent systems and service-oriented architectures, a significant incursion in industrial scenarios in use today is still missing; also, solutions using these concepts are not suffice in terms of re-configurability. In fact, re-configurability that can be defined as the ability of the system to dynamically change its configuration usually to respond to dynamic changes in its environment, e.g. a new production model, assumes a key role in the new generation of adaptive control systems providing the way to achieve a rapid and adaptive response to change, which is a key enabler of competitiveness.

In this context, there is a long path in the direction of reconfigurable and ubiquitous systems, able to integrate a network of production components to respond to the variability of production scenarios beyond those that were envisaged at design time. For this purpose, this kind of systems should integrate several complementary paradigms and technologies, combining the best they offer, following a 3-layer engineering framework, as illustrated in Fig. 1.

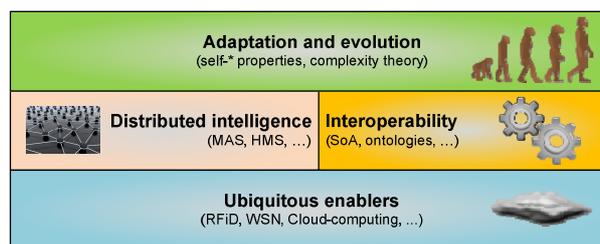


Fig. 1. A new vision for the engineering of complex adaptive systems

The core layer of this 3-layer framework is related to the **creation of a network of distributed and autonomous entities**, each one exhibiting intelligence and learning capabilities. This network is implemented using the MAS paradigm taking advantage

of the flexibility, robustness, scalability, adaptability, re-configurability and productivity they provide. The overall control system emerges from the interaction among these autonomous entities, which must be able to support efficiently the adaptation to the new unexpected scenarios and to respond to new business opportunities. The lacks in terms of interoperability exhibited by the MAS solutions can be overcome by combining the SOA principles, and especially by using the Web services technology. In fact, in spite of being based on the same concept of providing a distributed approach to the system, MAS and SOA present some important differences, namely in terms of autonomy and interoperability (see [9] for a deep study). These differences highlight the complementary aspects of the two paradigms, suggesting the benefits of combining them [9].

Aiming to support ubiquity, adaptation and re-configuration, two layers are glued to the core layer:

- A layer providing mechanisms to **create a ubiquitous environment**, supporting the dynamic reconfiguration and access to the services offered by the distributed agents.
- A layer providing biological inspired mechanisms to support the **adaptation and reconfiguration** of the system facing internal/external perturbations.

These layers provide a dynamic and evolvable system reconfiguration, which is one step ahead of traditional re-configurability, considering the evolution of the system and its components during its life-cycle, e.g., by offering new services or learning to differentiate normal from abnormal situations.

The details of these three layers will be discussed in the following sections.

3 Engineering Service-Oriented Multi-agent Systems

The effective development of distributed intelligent systems requires the combination of MAS and SOA paradigms, taking advantage of the best features they provide.

3.1 Creating Distributed Intelligence Using Multi-agent Systems

The traditional centralised and hierarchical approaches to complex control systems splits the overall problem into hierarchically dependent functions that operate within decreasing time-ranges, such as planning, scheduling and dispatching. This approach produces interesting results and near optimal solutions, but only when some assumptions are satisfied, for example, the absence of perturbations (e.g., machine breakdowns and urgent orders). However, industrial systems impose hard requirements with the frequent presence of deviations or perturbations, demanding a fast adaptation to the changing conditions. Consequently, this traditional approach rapidly becomes inefficient when the system must deal with such dynamic stochastic behaviour.

In this context, new intelligent and adaptive control systems must explore alternative control structures based on the decentralization of control functions over distributed entities, being the overall control system achieved by the cooperation among these entities. These systems can be seen as a network of **smart control components**

[10], controlling hardware devices (e.g., a sensor, a gripper or a robot), which can be reused whenever necessary. The smart control components comprise control and intelligence embedded in electronics devices, e.g., microcontrollers and programmable logic controllers, and communication capabilities to support the interaction with other components and the environment, as illustrated in Fig. 2.

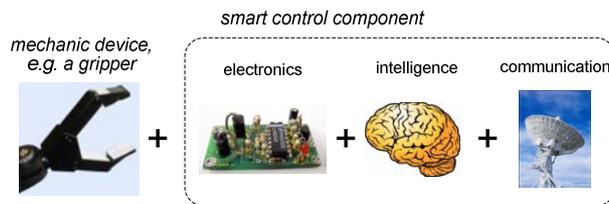


Fig. 2. Smart Control Component

Note that the smart control component concept recalls the *holon* concept, with the presence of the informational and physical parts [11]. In this sense, the thoughts provided by holonics can be applied in the development of such adaptive systems, namely the structural development of control systems through the encapsulation of functions and components in holons that are simultaneously self-contained wholes to their subordinated parts and dependent parts when seen from higher levels [12].

The implementation of this distributed intelligence network may use the MAS paradigm, which introduces a distributed control based on autonomous agents for the implementation of flexible, robust and reconfigurable plant control, characterised by the decentralisation of entities and functions [6], being the individual behaviours determined through the cooperation among agents, in opposition of being determined by a centralized mechanism. In this case, the intelligence module of the smart control component corresponds to an agent. According to [6], an agent is “*an autonomous component that represents physical or logical objects in the system, capable to act in order to achieve its goals, and being able to interact with other agents, when it doesn't possess knowledge and skills to reach alone its objectives*”. For example, in the manufacturing domain, an agent may represent a physical resource, such as a machine, a robot or a pallet, or a logical object, such as a scheduler or an order.

3.2 Distributed Entities Working Together Using Service-Oriented Principles

The smart control components, as parts of a complex and distributed system, are distributed autonomous entities which only have local knowledge and act to fulfil their own goals. The desired overall system is achieved by putting these smart control components working together, each one contributing with its behaviour (note that in such systems, the cooperation among agents is crucial). Following the concept of *swarm intelligence* [13], the global system behaviour of MAS emerges from a multiplicity of non-linear interactions among the individual entities. As the result, these systems have the capability to respond promptly to change, and differ from the conventional approaches due to their inherent capabilities to adapt to emergence without external intervention [3].

In distributed and heterogeneous environments it is important to guarantee the interoperability between the distributed entities or applications and to verify that the semantic content is preserved during the conversation between them. The solution to those problems requires the use of standard platforms that support transparent communication between distributed smart control components or applications. Ontologies [14] play a decisive role to support interoperability problems, being their ultimate goal the description, possibly without ambiguity, of a certain domain (i.e. the shared knowledge).

Complementarily to the use of ontologies, and since it is an unanswered problem in multi-agent systems, the interoperability can be overcome through the use of service-oriented principles. The SOA paradigm is a way of building distributed systems [15], which is based on the concept of providing and requesting services. A service is a piece of software that encapsulates the business/control logic or resource functionality of an entity that responds to a specific request. In such systems, the entities that want to offer their functionalities, encapsulate them as services and offer to the other entities by publishing them in a central repository. Using discovery mechanisms, service consumers find the services they need, and interact directly to get those services.

In the proposed approach, each smart control component encapsulates the functions that the physical device can perform as services, e.g. open or close the gripper. These services, that can be modified, added or removed (e.g. a new piece can be handled by a robot after the aggregation of a new gripper), are then exposed to be invoked by other smart control components that want to use them. In other words, the functionalities provided by the agents embedded in the smart control components are encapsulated as services and offered to other agents.

A main concern in service-based systems is how the services “play” together, emerging the concepts of orchestration and choreography. *Service orchestration* is the practice of sequencing and synchronizing the execution of services [16]. An orchestration engine implements the logic for the workflow-oriented execution and sequencing of services, and provides a high-level interface for the composed process. *Service choreography* is a complementary concept, which considers the rules that define the messages and interaction sequences that must occur to execute a given process through a particular service interface.

Orchestration mechanisms are of crucial importance to coordinate the complex and emergent behaviours of individual smart control components. These coordination mechanisms, that include orchestration engines for service composition, coordination and collaboration, must also consider interaction mechanisms that combine the component level with higher-levels of supervision to achieve cohesive distributed intelligent control [10]. An example of orchestration engines is the use of the Petri nets formalism [17], which is a mathematical and graphical oriented language for the design, specification, simulation and validation of complex event-driven systems, to describe the workflow representing the synchronization of services [18] as illustrated in Fig. 3. In this case, the orchestration engine has to interpret the workflow model expressed in the Petri nets model and execute it in real-time, synchronizing and controlling the whole process until the goal is reached. During this token-game process, the enabled transition must be detected, services associated with the enabled transition must be called and, after that, the workflow model has to be updated to reflect the actual state of the system. Looking at the example of Fig. 3, the services B.1 and C

are executed in parallel and after the execution of service A; the synchronization is required to start the execution of the service B.2 only after the complete execution of the two previous services.

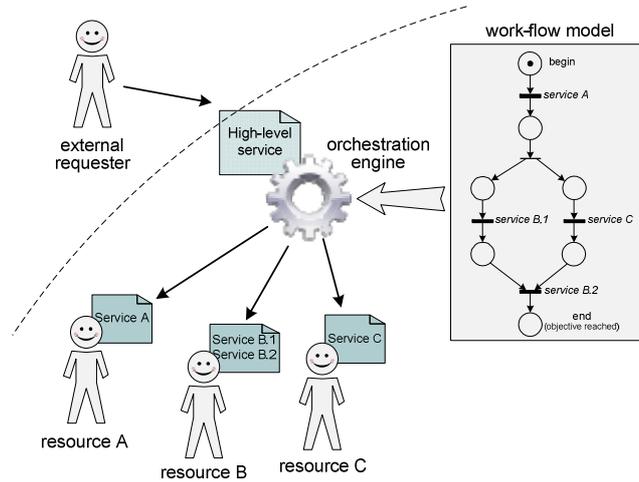


Fig. 3. Service Orchestration using Petri nets

The use of service-orientation principles supports the three dimensions of the collaboration manufacturing management (CMM) model [19] as shown in Fig. 4: it addresses the vertical enterprise integration by covering from the shop-floor level to the business level, the supply chain integration by supporting the interaction with suppliers and clients, and the life-cycle of a collaborative manufacturing system.

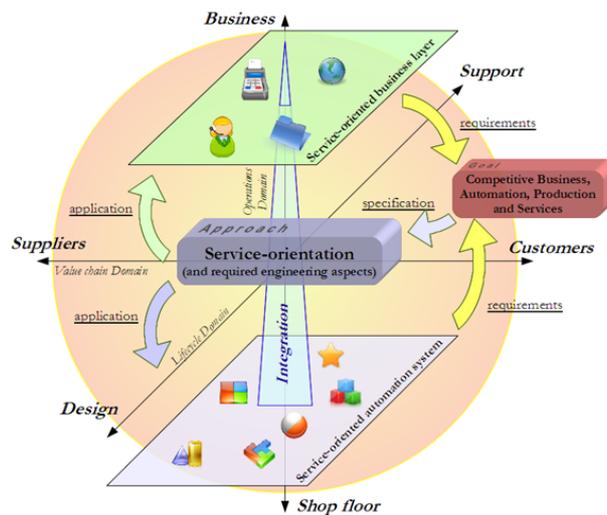


Fig. 4. Service-oriented ecosystem integrated to the CMM model [20]

This illustrates that the use of the service-orientation paradigm, e.g. implemented through Web services [21], enables the adoption of a unifying technology for all levels of the enterprise, from sensors and actuators to enterprise business processes [22].

The idea to combine complementary paradigms does not deplete with the consideration of the SOA principles. In the same manner, MAS can be integrated with other complementary technologies, e.g., IEC61131-3 and IEC 61499 approaches to implement the low-level control that is not addressed by the agents. This leads to the implementation of two levels of control: at the higher level, agents provide intelligence and adaptation, and at the lower level, IEC61499 guarantees real-time responsiveness.

3.3 Towards Service-Oriented Multi-agent System

In the sequence of the process previously described, the service-oriented principles can be integrated with multi-agent systems to enhance some functionality and to overcome some of its limitations, namely in terms of interoperability and IT-vertical integration. This suggestion is not new since services are already part of the agents' specification, e.g., in the Foundation for Intelligent Physical Agents (FIPA) specification [23], and agents are also present in standard documents of SOA methodologies, e.g., in the OASIS standard [24].

The service-oriented multi-agent systems (SoMAS) approach discussed in this work, and introduced by [25], is characterized by the use of a set of distributed autonomous and cooperative agents (embedded in smart control components) that use the SOA principles, i.e. oriented by the offer and request of services, in order to fulfil industrial and production systems goals. This approach is different from the traditional MAS mainly because agents are service-oriented, i.e. individual goals of agents may be complemented by services provided by other agents, and the internal functionalities of agents can be offered as services to others agents [25] (note that these service-oriented agents do not only share services as their major form of communication, but also complement their own goals with different types of external provided services).

Fig. 5 illustrates one SoMAS comprising devices representing conveyors (transporting pallets) and pallets. Each one of these devices has associated a service-oriented agent that is responsible of part of this environment (domain of autonomy).

The conveyor agent provides a service, called *transfer pallet*, which reflects its own functionality and skills. Therefore it has the ability to read the sensors, reason to execute its control behaviour and send commands to the actuators of the conveyor, to make it possible to transport pallets from the input to the output. This service is published in the Service Registry to be discovered by other agents representing devices, e.g., conveyors or pallets.

Other neighbour devices, e.g., a pallet agent that needs this transfer service to accomplish its goals, may request to the conveyor agent the service of transferring the pallet. However, to complete the service and also to respect global objectives, the conveyor must request an availability service from the next transport unit or workstation connected to its output. This can be seen as the form of collaboration among the service-oriented agents in the system.

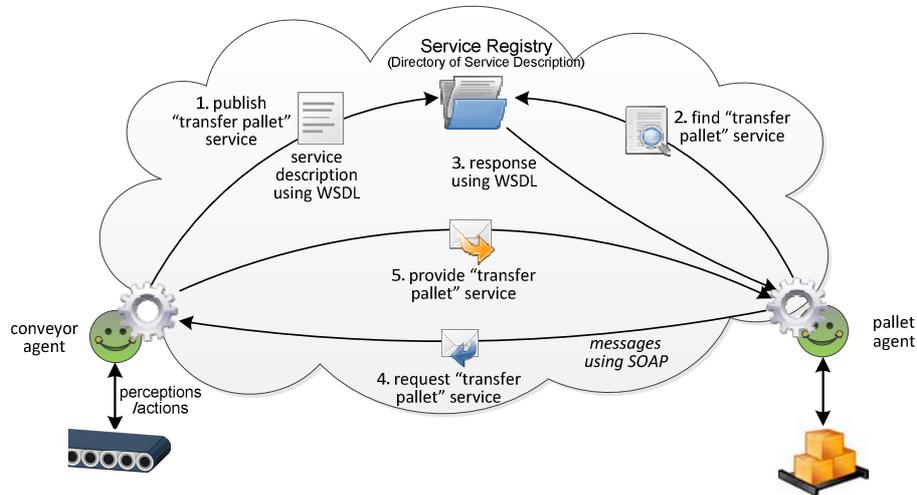


Fig. 5. Representation of a service-oriented agent and its environment

In conclusion, the adoption of the SoMAS satisfies the following requirements:

- Agents provide the control, intelligence and autonomy capabilities to manage the execution of a set of functionalities and activities.
- Resources, e.g., hardware devices and software modules, can be encapsulated as a service provider that acts like a bridge between the internal structure of the agent and the exposed interface to the outside world.
- Some services can be composed by other services, creating a levelled structure of services.

As result, the distributed nature of the architecture suggests the definition of interoperability functionalities based on service-oriented principles and the realization of efficient, flexible and robust overall plant control supported by the intelligence and autonomy provided by multi-agent systems.

4 Enhancing Service-Oriented Multi-agent Systems with Ubiquitous and Self-* Properties

Besides the combination of service-oriented principles with multi-agent systems, also of special interest is the articulation with other technological domains providing foundations to support distributed and ubiquitous environments, such as radio-frequency identification (RFiD) technology, wireless sensor networks (WSN), cloud and grid computing, and the consideration of biological inspired techniques namely emergence, self-organization and self-learning, to support adaptation and evolution at process control level.

4.1 Ubiquity Enablers

The main contribution of SoMAS is to offer a front-end layer with services encapsulating the functionalities provided by the agents, which act in the background. The achievement of a truly reconfigurable and adaptive system requires the existence of a ubiquitous environment where:

- The use of the RFID technology especially embedded as nanotechnology in products or resources will support the achievement of the concept of **intelligent product**. This emergent concept is related to products containing IT in the form of software, microchips and sensors, being able to collect and process information, generate knowledge and even provide reasoning capabilities.
- The use of WSN will support the **physical reconfiguration of the system** on the fly, i.e., the hardware devices can be added, removed or change their location without stopping and re-initializing the system or the other devices.
- The place where the services are placed is not important for the service requester. In this case, the consideration of the cloud computing will change the traditional way of thinking and will provide important characteristics in terms of capacity of storage and high-processing computing (HPC) facilities.

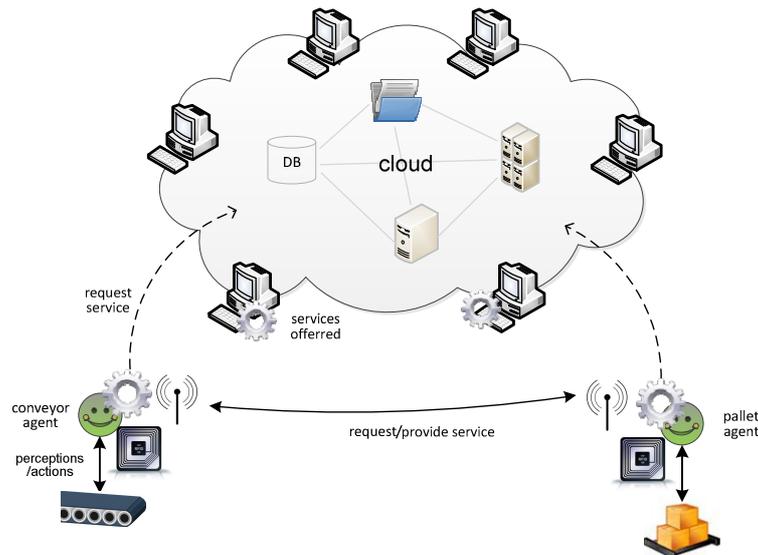


Fig. 6. Technology Enablers in SoMAS to Provide Ubiquity

Ideally, re-configuration should appear to users like “drag-and-drop” applications where complexity and details are handled by background services. The reconfiguration of any smart control component should be done on the fly, maintaining unchanged the behaviour of the entire system which should continue to run smoothly after the change. For this purpose, an environment based on cloud principles, where services can be distributed and offered, in a transparent way, should be considered.

4.2 Dynamic Adaptation and Reconfiguration

The SoMAS approach allows the achievement of significant advantages to address the current requirements, but it still lacks the capability to truly adapt and evolve to unexpected pressures from the environment, product fluctuation and internal/external disturbances.

Biology provides a plenty of simple mechanisms that solve complex problems, constituting suitable sources of inspiration to support the design of better solutions for adaptive and evolvable complex systems. Miller reinforces this idea by stating that “*the study of swarm intelligence is providing insights that can help humans to manage complex systems*” based on the idea that “*a single ant or bee isn't smart but their colonies are*” [26]. This can also be seen by the multiplicity of applications being currently developed using concepts and mechanisms inspired from the behaviour of species (see [27] for a detailed analysis).

Several self-* properties can be considered when developing such adaptive SoMAS, namely self-adaptation (i.e., the capability of an entity to change its behaviour depending on the external conditions), self-healing (i.e., the capacity to diagnose deviations from normal conditions and take proactive actions to normalize them and avoid service disruptions), self-learning (i.e., the capability to acquire new knowledge supporting the dynamic behaviour evolution) and particularly self-organization[28].

Self-organization is a theory found in biology that can be defined as the autonomous adaptation to the dynamic evolution of the environment [28]. Self-organizing systems do not follow a rigid and pre-defined organization; instead they evolve, without a central entity, through a non-linear and dynamic process with a constant optimization of the individuals' behaviour. In other words, self-organization can be seen as a set of rules and mechanisms that allow the dynamic self-formation of groups of distributed entities.

Stigmergy is probably the most well-known form of self-organization, involving an indirect coordination between entities, where the trace left in the environment stimulates the execution of a subsequent action, by the same or different entity. As an example, ants exchange information by depositing a chemical substance, known as pheromone, on their way back to the nest when they have found food. This odour can then be sensed by other ants, supporting the decision of the best route. Self-organization mechanisms can also be found outside the nature world, for example, the pattern formation in a Belousov-Zhabotinsky reaction or in physics where hot and cold molecules self-organize to create a flow.

The degree of efficiency of the self-organization capability is strongly dependent on how the learning mechanisms are implemented. In the design of self-organized systems the key issue is to define powerful intelligence mechanisms, not only including static intelligence mechanisms but also learning capabilities, that enable the system to improve its behaviour in the future as result of its experience. Particularly, learning mechanisms allow the evolution of the functionalities and behaviour of individual smart control components and consequently the evolution of the entire system, being critical to support the identification of re-configuration opportunities, as illustrated in Fig. 7. For this purpose, the embodied intelligence concept, associated to the

artificial life field [29], assumes a crucial role. This concept suggests that intelligence requires a body to interact with [30]; in this case, the intelligent behaviour emerges from the interaction of *brain* (i.e., the agent), *body* (i.e., the smart control component) and *environment*.

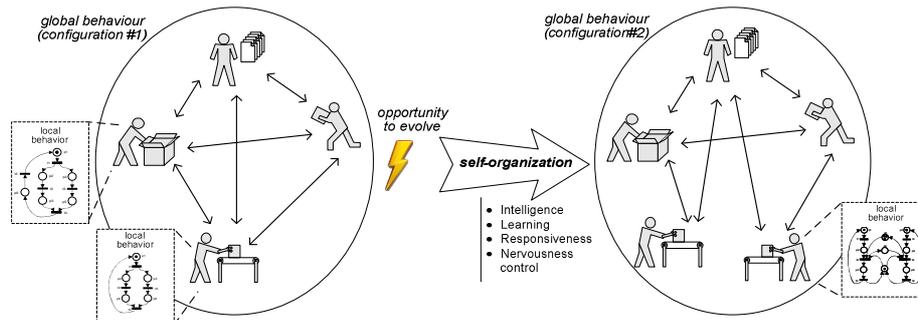


Fig. 7. Self-organized Service-oriented Multi-agent Systems

The dynamics of such self-organized and unpredictable systems can be understood with the support of the theory of complexity, which is a field of study trying to characterize complex systems. A complex system is a “*system composed of interconnected parts that as a whole exhibits one or more properties (behaviour among the possible properties) not obvious from the properties of the individual parts*” [31]. These complex systems have emergent properties that can’t be reduced to the behaviour of separated entities (i.e. the behaviour of separate entities does not explain the global behaviour of the system), as stated in [32]: “*the complexity of a system increases with the number of distinct components, the number of connections between them, the complexities of the components, and the complexities of the connections*”. This means that the analysis of these systems can’t be made by classical methods, e.g. Newtonian mechanics, which are essentially *reductionists*: divide the global problem into smaller problems, simpler to solve, and solve the global one by adding the small solutions. On contrary, in complex systems the global behaviour is more complex than the simple sum of all the small behaviours [33]. However, the reductionist approach can be used to understand the contribution of the individual parts keeping in mind the need to understand the unifying mechanisms of all the parts which will culminate in an emergent behaviour.

During the reconfiguration process, the objective is to maintain the system behaviour predictable and stable but since these systems are non-linear and unpredictable, some instability can appear as the result of not properly synchronized evolution processes. Additionally, the butterfly effect may appear in such systems, where a small change at one place in a deterministic non-linear system may cause large changes. The solution is to design mechanisms ensuring that desired properties and behaviours will emerge, whereas not expected and not desired properties and behaviours will not.

Some attention should be devoted during the application of the biological inspiration to solve a determined problem. It may happen that not all of the biological inspired behaviour could be important to solve a given problem. The researcher must

take into consideration what is really needed, discarding what is not important. This is a crucial issue since the imitation of unnecessary behaviours complicates the implementation and could enforce undesired performances of the solution. Some considerations should also be made regarding the differences between the nature and engineering worlds. In nature, there is time and space to recover after failures, which means that if something isn't made right at first time there are always more opportunities to get it right. On other hand, in the engineering world things must be done right at the first time and failures must be avoided at all cost. Another important difference is that the main goal in nature is to guarantee the species survival while engineering has a multiplicity of very specific goals (e.g. costs reduction, ensure quality and customization).

5 Adoption by Industrial Players

An important question is concerned with the possible acceptance and adoption of this 3-layer perspective for the development of such cyber-physical systems by the industrial players. The past experience in applying individually these concepts, for example MAS, was not a complete successful story. In fact, the analysis of the industrial applications of agent-based solutions reported in the literature (see [6] and [7] for more details) allows extracting the following conclusions [34]:

- There is a relatively small adoption of agents in industry, being the implemented applications limited in terms of functionality.
- The developed solutions address mainly the high-level control or the pure software systems (e.g., the electronic commerce), and not those imposing (soft or hard) real time constraints or the connection to physical devices.
- There is a reduced enthusiasm from both the technology providers and the industry end-users companies.

The reasons for this weak adoption in industry were already widely discussed in the literature [6; 8]. Briefly, the main road-blockers are the required initial investment, the need to adopt the distributed thinking, the interoperability in distributed heterogeneous systems, the missing standardization, the real-time constraints and the missing technology maturity. Additionally, the presence of several “fuzzy” concepts leading to decentralization and unpredictability, such as emergence, self-organization, learning and chaos, are not completely assimilated by industrial players.

Having in mind that the complete adoption of such concepts and technologies is a long and difficult road, where the current “nice-to-have” approach should lead to the “must-to-have” approach in the future [35], the challenge is to learn from the experience and try to understand the reasons for this situation, implementing actions that may contribute for a wider acceptance of these concepts by industrial players. The main actions to be considered are the convincement of the industry people, achievement of standardization and fulfilment of resilience and security issues.

The **conviction of the industry people** of the benefits of using this approach, e.g., by providing demonstrators running in industry that shows the maturity, flexibility

and robustness of self-organized SoMAS solutions, is probably the major and most important action to be performed. This action will allow industrial companies to “believe” in these concepts and technology. A promising perspective is the interest and efforts devoted by several software developers companies, e.g., Magenta Technology, Smart Solutions, NuTech Solutions and Whitestein Technologies, and by several automation technology providers, e.g. Rockwell Automation, Schneider Electric and Siemens, to apply these concepts in their business solutions.

The **standardization** is pointed out by industry as a major challenge for the industrial acceptance of the agent technology, since standards may affect the development of industrial MAS solutions. Examples of standards that can be related to self-organised SoMAS solutions are ISA-95, IEC 61131-3 and IEC 61499, FIPA (Foundation for Intelligent Physical Agents), and Semantics and ontology standards.

An important indicator that this issue is currently being taken into consideration is the importance that the standardization issues start to have in R&D projects. As an example, the European FP7 ARUM (Adaptive Production Management) project includes a complete task devoted to this issue.

In cyber-physical systems exhibiting distributed environments, **resilience** and **security** are critical issues in industrial applications and systems, especially those that are mission critical. The resiliency approaches often aim to adapt to highly non-linear, often ill-defined disturbances, which are frequent in such cyber-physical systems. The security in the data exchange and sharing is crucial in non-trustful systems and networks, e.g. supply chain or virtual organization processes.

The implementation of these actions may lead to the development of more powerful self-organized SoMAS solutions that may be better accepted by industry.

6 Conclusions

This paper discusses the advantages of combining multi-agent systems with complementary paradigms, and particularly service-oriented architectures, to design more powerful adaptive control systems, based on the decentralization of control functions over distributed autonomous and cooperative entities. Additionally, aiming to reach a truly self-adaptive and evolvable system, the paper discusses how to enrich the system designed using the previous paradigms with ubiquitous enablers, such as cloud computing and WSN, and biological inspired techniques namely self-organization, taking into consideration that in nature very complex and adaptive systems are implemented by using very simple behaviours and mechanisms.

In fact, the incorporation of self-* mechanisms, and particularly self-organization in SoMAS can greatly contribute to increase the system performance, flexibility, robustness, adaptation and re-configurability, taking advantage of the capabilities provided by these techniques. To illustrate this concept, consider the following analogy: imagine the latest Ferrari F1 car equipped with the state-of-the-art tools and techniques to have a wonderful full drive experience, which can be tested by two different drivers: a typical daily driver and a F1 world champion. Naturally, only the F1 world champion driver is able to push the car into its limits extracting every available horse

power, making every curve more exciting and continuously pushing the car, because he has the required knowledge and experience to do that. The F1 world champion driver can be compared to the self-organization mechanisms in the sense that can push the system to its limits taking advantage of the full capabilities provided by the SoMAS infrastructure.

References

1. Getov, V.: Smart Computational Grids: Infrastructure for Tackling Global Integration Challenges. News Release. IEEE (August 15, 2012)
2. Lee, E.: Cyber Physical Systems: Design Challenges. Technical Report No. UCB/EECS-2008-8, University of California, Berkeley (2008)
3. Wooldridge, M.: An Introduction to Multi-Agent Systems. John Wiley & Sons (2002)
4. James, F., Smit, H.: Service-Oriented Paradigms in Industrial Automation. *IEEE Transactions on Industrial Informatics* 1(1), 62–70 (2005)
5. Vyatkin, V.: IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design, 2nd edn. ISA (2011)
6. Leitão, P.: Agent-based Distributed Manufacturing Control: A State-of-the-art Survey. *Engineering Applications of Artificial Intelligence* 22(7), 979–991 (2009)
7. Monostori, L., Váncza, J., Kumara, S.: Agent-Based Systems for Manufacturing. *Annals of the CIRP* 55(2), 697–720 (2006)
8. Marik, V., McFarlane, D.: Industrial Adoption of Agent-Based Technologies. *IEEE Intelligent Systems* 20(1), 27–35 (2005)
9. Ribeiro, L., Barata, J., Mendes, P.: MAS and SOA: Complementary Automation Paradigms. In: Azevedo, A. (ed.) *Innovation in Manufacturing Networks*. IFIP, vol. 266, pp. 259–268. Springer, Boston (2008)
10. Leitão, P., Mendes, J., Colombo, A.W., Restivo, F.: Reconfigurable Production Control Systems: Beyond ADACOR. In: *Proceedings of the IFAC Intelligent Manufacturing Systems Workshop (IMS 2007)*, Alicante (2007)
11. Winkler, M., Mey, M.: *Holonic Manufacturing Systems*. European Production Engineering
12. Koestler, A.: *The Ghost in the Machine*. Arkana Books, London (1969)
13. Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence: from Natural to Artificial Systems*. Oxford University Press (2009)
14. Gruber, T.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies* 43(5/6), 907–928 (1995)
15. Ross-Talbot, S.: *Orchestration and Choreography: Standards, Tools and Technologies for Distributed Workflows*. NETTAB Workshop - Workflows Management: New Abilities for the Biological Information Overflow (2005)
16. Jammes, F., Smit, H., Martinez Lastra, J.L., Delamer, I.: Orchestration of Service-Oriented Manufacturing Processes. In: *Proceedings of the 10th IEEE International Conference on Emergent Technologies for Factory Automation (ETFA 2005)*, vol. 1, pp. 617–624 (2005)
17. Murata, T.: Petri Nets: Properties. *IEEE Analysis and Applications* 77(4), 541–580 (1989)
18. Mendes, J.M., Leitão, P., Restivo, F., Colombo, A.W.: Process Optimization of Service-Oriented Automation Devices Based on Petri Nets. In: *Proceedings of the 8th IEEE International Conference on Industrial Informatics (INDIN 2010)*, pp. 274–279 (2010)
19. Gorbach, G., Nick, R.: *Collaborative Manufacturing Management Strategies*. White paper, ARC Advisory Group (2002)

20. Mendes, J.M.: Engineering Framework for Service-oriented Automation Systems. PhD Thesis, University of Porto (2011)
21. ***: World Wide Web Consortium (W3C). Web Services Glossary (2011), <http://www.w3.org/TR/ws-gloss/>
22. Bepplerling, A., Mendes, J.M., Colombo, A.W., Schoop, R., Aspragathos, A.: A Framework for Development and Implementation of Web service-Based Intelligent Autonomous Mechatronics Components. In: Proceedings of the IEEE International Conference on Industrial Informatics, Singapore, pp. 341–347 (2011)
23. ***: FIPA Abstract Architecture Specification. Standard of the Foundation for Intelligent Physical Agents (2002), <http://www.fipa.org/specs/fipa00001>
24. ***: Reference Model for Service Oriented Architecture 1.0. OASIS Standard (October 12, 2006), <http://docs.oasis-open.org/soa-rm/v1.0> (2006)
25. Mendes, J.M., Leitão, P., Restivo, F., Colombo, A.W.: Service-Oriented Agents for Collaborative Industrial Automation and Production Systems. In: Mařík, V., Strasser, T., Zoitl, A. (eds.) HoloMAS 2009. LNCS, vol. 5696, pp. 13–24. Springer, Heidelberg (2009)
26. Miller, P.: The Genius of Swarms. National Geographic (2007)
27. Leitão, P., Barbosa, J., Trentesaux, D.: Bio-inspired Multi-Agent Systems for Reconfigurable Manufacturing Systems. Engineering Applications of Artificial Intelligence 25(5), 934–944 (2012)
28. Leitão, P.: Holonic Rationale and Self-organization on Design of Complex Evolvable Systems. In: Mařík, V., Strasser, T., Zoitl, A. (eds.) HoloMAS 2009. LNCS, vol. 5696, pp. 1–12. Springer, Heidelberg (2009)
29. Adami, C.: Introduction to Artificial Life. Springer (1998)
30. Pfeifer, R., Scheier, C.: Understanding Intelligence. MIT, Cambridge (2001)
31. Joslyn, C., Rocha, L.: Towards semiotic agent-based models of socio-technical organizations. In: Proceedings of the AI, Simulation and Planning in High Autonomy Systems (AIS 2000) Conference, pp. 70–79 (2000)
32. Richardson, K.: Managing Organizational Complexity: Philosophy, Theory and Application. Information Age Publishing (2005)
33. Holland, J.: Emergence: from Chaos to Order. Oxford University Press (1998)
34. Leitão, P., Marik, V., Vrba, P.: Past, Present, and Future of Industrial Agent Applications. Accepted to be Published in IEEE Transactions on Industrial Informatics (2012)
35. Li, K., Wang, C., Ghenniwa, H., Shen, W.: A Service-Oriented Approach for Real Time Shop Floor Monitoring and Control. In: Shen, W. (ed.) Information Technology for Balanced Manufacturing Systems. IFIP, vol. 220, pp. 395–404. Springer, Boston (2006)