Self-interested Service-Oriented Agents Based on Trust and QoS for Dynamic Reconfiguration

Nelson Rodrigues, Paulo Leitão and Eugénio Oliveira

Abstract Progressively increasing complexity of dynamic environments, in which services and applications are demanded by potential clients, requires a high level of reconfiguration of the offer to better match that ever changing demand. In particular, the dynamic change of the client's needs, leading to higher exigency, may require a smart and flexible automatic composition of more elementary services. By leveraging the service-oriented architectures and multi-agent system benefits, the paper proposes a method to explore the flexibility of the decision support for the services' reconfiguration based on several pillars, such as trust, reputation and QoS models, which allows the selection based on measuring the expected performance of the agents. Preliminary experimental results, extracted from a real case scenario, allow highlighting the benefits of the proposed distributed and flexible solution to balance the workload of service providers in a simple and fast manner. The proposed solution includes the agents' intelligent decision-making capability to dynamically and autonomously change services selection on the fly, towards more trustworthy services with better quality when unexpected events happen, e.g. broken machines. We then propose the use of competitive self-interested agents to provide services that best suits to the client through dynamic service composition.

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, 5301-857 Bragança, Portugal

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

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-agent Manufacturing*,
Studies in Computational Intelligence 594, DOI 10.1007/978-3-319-15159-5_20

Keywords Multi-agent systems · Service composition · Self-organization

1 Introduction

The manufacturing industry is facing self-organization and evolution, the implementation of shop floor flexibility and reconfiguration being regarded as some of the many global challenges of the upcoming decade [1]. The literature is full of good examples regarding the benefits of the Service-Oriented Architecture (SOA), seen as an excellent solution to face the many current industry challenges, namely providing interoperability in heterogeneous systems and adaptability to condition changes, allowing companies to save time and money by simplifying the execution of complex tasks to be carried out. As a consequence, a big effort is being developed to embed intelligence and autonomy in the services' discovery, aggregation and composition processes, which can be achieved by using Multi-agent Systems (MAS) combined with SOA [2]. In addition, the dynamic and automatic reconfiguration of the distributed system, e.g., by adapting or creating new services offered by the several autonomous agents to face the new identified requirements, needs to be studied. Self-organization in MAS is being explored aiming to achieve, in dynamic and open environments, more trustworthy and automatic reconfiguration services. Assuring modular capabilities, a system is composed of components that are organized in a variety of configurations, flexible enough to execute different services and intelligent to reconfigure the services in an automatic manner [3].

This paper proposes a model to design and evaluate, at run-time, several service composition hypotheses, allowing selecting the best proposals that will support the agent's self-organization and the improvement of its utility. The proposed hypothesis increases the service composition quality by appropriately selecting the best service providers based on several non-functional requirements (e.g. service performance, cost, availability, besides the response time), usually referred as Quality of Service (QoS) [4]. The agent's ability to interact and choose the best services is based on certain Key Performance Indicators (KPI) that express the confidence in the services' providers, such as quality, reputation and trust. In a competitive mode, the agents adjust their own KPIs to improve the possibility to become credible candidates for selection, increasing the utility of each client agent. The proposed concepts were applied to a case study and the preliminary results show how important may be the dynamic service allocation through competitive agents, allowing scheduling a production plan by balancing the task effort.

The rest of the paper is organized as follows: Sect. 2 presents the literature review and Sect. 3 introduces the proposed model. Section 4 describes the experimental case study and performs a critical analysis of the achieved results. Finally, Sect. 5 wraps up the paper by stating the conclusions.

2 Literature Review

The importance given to the selection of the optimal services has been widely studied in the literature [5], being strongly related to the selection of services in a centralized repository. Initially, this selection process was carried out in a manual manner, but lately, some efforts are driven to automating the discovery and composition phases. The optimum process composition, considering the optimization of the workflow composition by QoS features, is not new [6]. The nature of workflows design is devoted by complex systems concepts, where MAS represents a suitable paradigm to implement and facilitate such complexity, by subdividing the problem space in a distributed, flexible and autonomous manner. Some authors analyse the flexible workflow topics [6, 7], proposing methodologies to coordinate the workflows and raising also the service-oriented computing to support inter-organizational workflows. Buhler and Vidal refer that, theoretically, such adaptation should appear regarding the monitoring of self-* properties [8], but the authors do not provide details on the trigger learning module. Mendes et al. [9] propose the service composition in the industrial domain, formal specified by Petri Nets models, however no learning mechanisms are explored. Our work takes into consideration the recent proposal of Vogel and Giese [10] that propose a model-driven for self-adaption based on feedback loops and the selection of the most promising agent [11]. Essentially, there is a lack of implementing approaches regarding self-organization features that consider distributed MAS [3] selecting the entity that provides the best service. In fact, the literature review does not offer expressive enough detail and results in the creation of such systems. This lack of detail offers a research opportunity to adapt self-interested agents, based on self-organization to actively re-configure service's features on the fly, improving the composition concerned.

3 Proposed Approach

In emergent and volatile environments, systems need to quickly adapt to changes or unexpected disruptions [12], e.g. changes in operating conditions or product demands.

3.1 Combining Multi-agent Systems and Service-Orientation

To deal with this kind of scenarios, the proposed architecture combines MAS and SOA paradigms to support an easy decentralization and distribution of decision-making entities, allowing that autonomous and self-* agents adapt dynamically and in a distributed manner their behaviours in due time (see Fig. 1). The SOA principles provide several features, such as standardized service contracts, loose coupling,

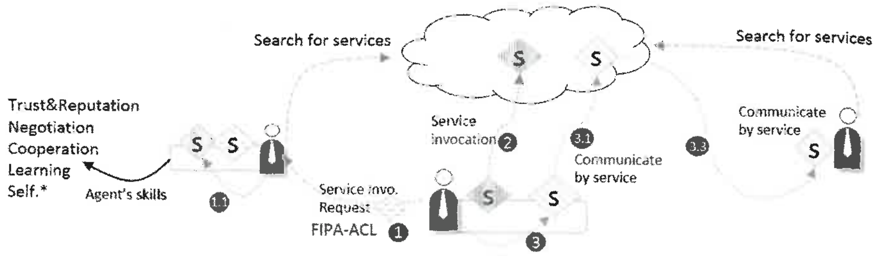

Fig. 1 Combining the interoperability of SOA and the intelligence and autonomy by MAS

service abstraction and service reusability, among others, allowing designing complex systems based on the offer and consumption of those services, each one encapsulating the functionalities of an intelligent system or agent. The agents communicate among themselves using FIPA-ACL messages to request a service execution (1), or by invoking directly the service (2), or lastly, by invoking a service that encapsulates others services (3). In this way, services provided by the agents can be consumed.

Each agent has the skills (e.g., trust mechanisms or learning) that by default are not publicly exposed. Depending on the competitiveness or cooperation of the agents system, some skills are mapped as services and others left as private. In the proposed MAS, the following agents were included:

- *Provider agent* (PA) having the capability to provide one or more services that can be performed and correspond to its skills. The PA is autonomous to dynamically change the offered services as well as the associated conditions, e.g., price and QoS, and tries to offer its services to be used as much as possible, inside its own limitations.
- *Consumer agent* (CA) corresponding to the entity that requests a service that is required to execute its workflow. The CA is autonomous to decide which service best suits its particular needs. During the negotiation with the PAs, the CA aims to increase its utility, choosing the service needed with lower price and higher trust, reputation and QoS value by default. Moreover, each CA has an internal database to collect the feedback knowledge about the QoS, trust, and prices about several services of a specific PA.
- *Discover agent* (DA) providing self-discovering mechanisms that are required to find potential services relevant to the consumer, i.e. the agent can reason of the services' skills (e.g. trying to create possible coalitions between services).
- *Workflow agent* (WA) having the responsibility to monitor the quality of the available services and to dynamically create the best composition of services. For this purpose, CA interact with the WA expecting to receive the best composed planning scenarios, which are ranked by the quality, price and reliability of the composition within the consumer constraints.
- *Ontology agent* (OntA) having the capability to clarify possible misunderstandings during the agents' interaction for the negotiation and cooperation processes, e.g. applying semantic-based mechanisms to translate concepts.

- *Reputation* agent (RA) and *observer agent* (OA) work in a centralized manner by gathering the results of the services' execution, being able to provide advising based on their expertise. The RA collects a feedback value after each executed service and the OA collects the information after the execution of each process plan. These agents support their decisions based on prediction algorithms working upon the past events.

Each agent is autonomous, being able to choose which task to perform, e.g. an agent may refuse the execution of a service if it perceives that a service will jeopardize its internal functionality. The agent's roles also allow the system to dynamically self-organize to evolve with more confidence becoming more robust.

3.2 Service Selection Mechanisms

The analysis of the performance of PAs by CA agents must consider several KPI and thus, a multi-criteria function formalized by a Multi-Attribute Utility Theory (MAUT) to maximize the agent's utility is used. From the several possible criteria to evaluate possible partners for a business agreement, the trust, reputation, QoS and price (with different weights) were selected. Furthermore, it is important to specify the attributes that are associated with the services and agents. In Table 1 service variables were built based on both the models presented in [4] for QoS,

Table 1 Representation of the agent and services variables

Name and equation	Description	Actors
QoS availability $f(\phi) = \frac{\sum \lambda}{\sum \lambda + \sum \psi} * 100$	Ratio of the service uptime of time period, λ standing for service uptime and ψ for the service down time	Service
QoS response time $f(\theta) = \delta - \rho$	Performance of a service. Given by the difference between conclusion time δ, and ρ the request time	Service
QoS throughput $f(\eta) = \frac{\sum \gamma}{\sum \tau}$	Provider performance index. Given by the maximum number of services to process in a unit of time where γ stands for the complete request and τ for the unit time	Service
QoS processing time	Cost associated to an execution time	Service
Price	Cost function	Service
Trust Trt(Xtr, Ytr, Cxt, G, t)	Trust value of the agent in performing a specific service, where *Xtr* represents the trustor, *Yte* means the trustee (the one under assessment), in a context *Cxt*, for a specific goal *G* at the time *t*	Agent
Reputation Rp(Tp, Xtr, Yte, Cxt, G, t)	Trust value of the agent, where tp = {tp1, ..., tpk} is a set of recommendations, given by a third-party recommender Tp about a trustor agent Xtr	Agent

and Jonker and Treur [13] for the trust model. Trust and reputation are important criteria to measure the confidence and the risk that may be implicit in a future bilateral contract.

Moreover, and according to each specific situation, the weights of the different criteria can be customized. For example, if the client prefers a reliable product with higher quality than the cheaper product, then the WA agent must select the best services based on trust and QoS. From the PA side, the goal is always to increase its utility, by selecting the best strategy to maximize the revenue.

The PA agent that sooner recognizes the needs of consumers, early adapts to eventually be the selected provider. This competitive and adaptive approach allows to dynamically scheduling the tasks without using a complex scheduling algorithm, allowing at the same time to balance the task efforts for each producer. However, this heuristic-based methodology raises relevant questions: how to discover opportunities to adapt and how, at the same time, should be the dynamic selection of composed services performed. Each agent must explore the opportunities to adapt, e.g., when the agent is not performing any service or when the agent's utility is changing, particularly having a decreasing trend. These are possible opportunities considered in the case study.

4 Case Study

Aiming to validate the work, the proposed approach was implemented for the flexible manufacturing system "AIP PRIMECA" [14], which is composed of several stations connected through conveyors. Each station performs a set of operations (i.e. services) but only one single operation at a time. The catalogue of products comprises several distinct products, namely "B", "E", "L", "T", "A", "I" and "P", each one containing a production plan (workflow) that defines the sequence of operations for each product. For example, the production of the product "B" requires the execution of the *Plate_Loading* service, three times the *Axis* service, twice the *r_comp* service, followed by the sequence *l_comp*, *Screw_comp* and *Inspection* services and finally the *Product_unloading* service. These services can be performed in different stations; however, each station has different performances (see [14] for further analysis).

4.1 Case Study Agentification

The agent-based model for the production system case study was implemented using JADE (Java Agent Development Framework) [15]. In particular, the behaviours of PA, CA, DA, RA, WA and OA agents were implemented according to the proposed architecture, as well the interaction patterns among these agents.

Regarding the implementation phase, a scenario with two different batches of products was considered:

- #a: 2 batches of "BELT" products (comprising the "B", "E", "L", "T" sub-products), totalizing the execution of eight products.
- #b: 1 batch of products "AIP" (comprising "A", "I", "P"), totalizing the execution of 3 products.

The system starts by initializing 7 PA to represent the stations disposed in the production system, 1 OA to supervise the production system activities, 1 RA, 1 OntA and finally, 11 CA agents, one for each product instance. To simplify the case study, only one criterion, i.e. the price, will be shown for evaluating the producer's proposals.

4.2 Analysis of the Experimental Results

The case study production system is sufficiently flexible for the agents to extract the benefits of it, e.g. the same service can be offered by several PAs with different features, e.g. Machine2 and Machine3 are PA agents that provide the service *r_comp*.

The experimental tests consider a simulated dataset running in static and dynamic environments. In the first round, the utility of agents follows a static behaviour in relation to the price, e.g. PAs never change the price of a particular service. In the second round, the prices were dynamically adjusted according to the profit. Figure 2 depicts the difference of PA utilities for the first and second round. In the first round, the Machine2 PA had executed the *r_comp* service with a utility of 8 units (i.e. an indication of the cost) during 50 iterations (see left graph of Fig. 2). At this time, Machine3 PA was only able to perform the *Axis* service. In the second round, the Machine3 PA has changed the service's characteristics of the service *r_comp*, which has increased its utility (in this case changing only the price; see right graph of Fig. 2).

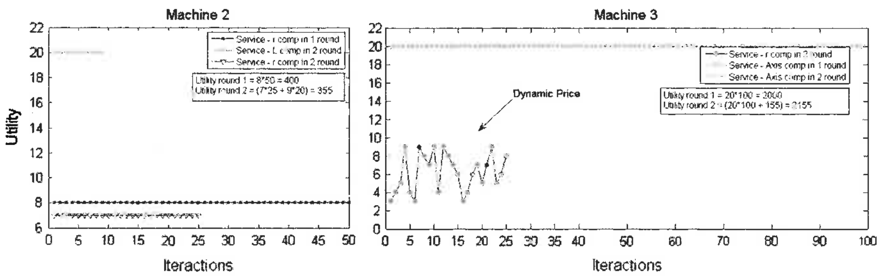

Fig. 2 Utility of services provided by Machine2 and Machine3

Table 2 Achieved Cmax using fixed or dynamic plans with/without perturbations

Scenario	Cmax (m:s) x = Trust (0.2) y = QoS(0.3)		Cmax (m:s) x = Trust (0.4) y = QoS(0.1)	
	Fixed	Dynamic	Fixed	Dynamic
Without perturbations	32:23	33:26	32:31	32:50
With perturbations	47:26	45:48	45:41	45:21

The reason for the dynamic behaviour in the Machine3 PA, illustrated in the right graph of Fig. 2, is related to the fact that sometimes the service *r_comp* is sold too cheap and the PA agent pretends to increase its utility, thus increasing the price. But in other hand, sometimes the price of *r_comp* service offered by Machine3 PA is set too high and consequently is not being sold (being instead selected the service offered by the Machine2 PA). Again, to increase its utility the Machine3 PA has to reduce the price, and consequently, this competitive behaviour offers opportunities for other PA agents to perform services. CA agents can choose the provider of the services, instead of a rigid plan, allowing the increase of the utility of the services and agents in the second round. The PA can observe and identify strategies being carried on in the environment, being capable to explore any market-based strategies.

During the simulation, the criterion to be optimized was Cmax, representing the amount of time necessary to complete a group of jobs. The weights for the multi-criteria function used in the service selection are: 0.3 for the *price*, 0.2 for the *Rep*, *x* for the *Trust* and *y* for QoS (see *x* and *y* values in Table 2). The experimental tests also considered the existence of perturbations, e.g. delays in the process execution, which will provoke a lack of fulfilling the commitment agreed in the negotiation (accordingly, the CA considers the failure of the commitment as a negative reward of a particular provider). The achieved results are illustrated in minutes and seconds in Table 2.

Analysing the impact of the multi-attribute function with a trust weight of 0.2 and without perturbations, the fixed approaches achieve a lower Cmax, which is expected since there are no errors and the plan is executed without deviations. In the dynamic approach, the achieved Cmax is considerably higher for each negotiation protocol, being more prominent when the production batch is bigger. Obviously, in the scenarios with the presence of perturbations, the Cmax is higher. Additionally, it can be realized that the fixed approach in the presence of perturbations has higher Cmax values than the dynamic one, due to the flexibility provided by the dynamic approach to detect, in a pro-active manner, that a station has some kind of perturbation, and adapt properly.

The same tendencies were achieved using a trust value of 0.4. Note that PA agents with better confidence do not guarantee that the service has higher quality. In some cases, the Cmax is higher if the confidence increases too much. Thus, the agent's capabilities to select and evaluate the correct providers infer the final product quality.

5 Conclusions

The progressively increasing complexity of dynamic environments, in which services and applications are demanded by potential clients, requires a high level of reconfiguration of the offer to better match that ever changing demand. To address this challenge, a service-oriented multi-agent system approach was proposed, to intelligently and dynamically select the most appropriate services provided by the reliable entities to increase the confidence and quality of the needed service composition. This paper explores the challenges of selecting services considering self-adaption capabilities, in order to improve the services availability and quality. The heuristic-based game played by the agents for the services selection based on trust, reputation, QoS and price, allows balancing the workload of the providers in a simple and fast manner. The proposed self-oriented agent-based system was applied to a simple production system case study aiming to validate its applicability. The achieved results lead to the conclusion that the system performance was improved along the following lines:

(i) Reducing the impact (down-times) of broken processes.
(ii) Reacting faster and accurately to condition changes, offering products with higher quality and without spending time implementing a dedicate schedule function.

In conclusion, the results convinced us to extend our approach to other domains, namely considering a distributed reputation model, and negotiation strategies to improve the system's intelligent decision capability.

References

1. Filos, E.: Manufacturing innovation and horizon 2020. In: Digital Product and Process Development Systems SE—1, vol. 411, pp. 1–10. Springer, Berlin (2013)
2. Huhns, M.N.: Agents as web services. Int. Comput. IEEE **6**(4), 93–95 (2002)
3. Rodrigues, N., Oliveira, E., Leitão, P.: Self-organization Combining Incentives and Risk Management for a Dynamic Service-Oriented Multi-agent System, in Technological Innovation for Collective Awareness Systems, pp. 101–108. Springer, Berlin (2014)
4. Ran, S.: A model for web services discovery with QoS. SIGecom Exch. **4**(1), 1–10 (2003)
5. Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K.: Quality of service for workflows and web service processes. Web Semant. **1**, 281–308 (2004)
6. Liu, Z.-Z., Xue, X., Shen, J., Li, W.-R.: Web service dynamic composition based on decomposition of global QoS constraints. Int. J. Adv. Manuf. Technol. **69**(9–12), 2247–2260 (2013)
7. Madhusudan, T.: An agent-based approach for coordinating product design workflows. Comput. Ind. **56**(3), 235–259 (2005)
8. Buhler, P.A., Vidal, J.: Towards adaptive workflow enactment using multiagent systems. Inf. Technol. Manag. J. **6**, 61–87 (2004)
9. Mendes, J.M., Leitao, P., Restivo, F., Colombo, A.W.: Composition of petri nets models in service-oriented industrial automation. In: 8th IEEE International Conference on Industrial Informatics (INDIN), pp. 578–583 (2010)

10. Vogel, T., Giese, H.: Model-driven engineering of self-adaptive software with EUREMA. ACM Trans. Auton. Adapt. Syst. **8**(4), 18:1–18:33 (2014)
11. Urbano, J., Rocha, A.P., Oliveira, E.: Trust-based selection of partners. In: E-Commerce Web Technologies, pp. 221–232. Springer (2011)
12. Rodrigues, N., Leitão, P., Oliveira, E.: Dynamic Composition of Service Oriented Multi-agent System in Self-organized Environments. In Proceedings of the 2014 Workshop on Intelligent Agents and Technologies for Socially Interconnected Systems (IAT4SIS '14), Czech Republic, Prague (2014). doi:10.1145/2655985.2655990
13. Jonker, C.M., Treur, J.: Formal analysis of models for the dynamics of trust based on experiences. In: Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: MultiAgent System Engineering, pp. 221–231 (1999)
14. 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)
15. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE: a software framework for developing multi-agent applications. Lessons learned. Inf. Softw. Technol. **50**(1–2), 10–21 (2008)

Volatile Knowledge to Improve the Self-adaptation of Autonomous Shuttles in Flexible Job Shop Manufacturing System

Emmanuel Adam, Damien Trentesaux and René Mandiau

Abstract It is well known now that MAS are particularly adapted to deal with distributed and dynamic environment. The management of business workflow, data flow or flexible job shop manufacturing systems is typically a good application field for them. This kind of application requires flexibility to face with changes on the network. In the context of FMS, where products and resources entities can be seen as active, this paper presents an application of the volatile knowledge concept to the management of a flexible assembly cell. We illustrate our proposition on an emulator of the flexible assembly cell in our university.

Keywords Volatile knowledge · Flexible job shop manufacturing systems · Multi-agent system

1 Introduction

To be competitive, manufacturing industries adapt to changing conditions imposed by the market. The greater variety of products, the possible large fluctuations in demand, the shorter lifecycle of products expressed by a higher dynamics of new products, and the increased customer expectations in terms of quality and delivery time are challenges that manufacturing companies have to deal with to remain competitive.

E. Adam (✉) · D. Trentesaux · R. Mandiau
LAMIH Lab., UMR CNRS 8201, UVHC, 59313 Valenciennes, France
e-mail: emmanuel.adam@univ-valenciennes.fr

D. Trentesaux
e-mail: damien.trentesaux@univ-valenciennes.fr

R. Mandiau
e-mail: rene.mandiau@univ-valenciennes.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-agent Manufacturing*, Studies in Computational Intelligence 594, DOI 10.1007/978-3-319-15159-5_21