



HAL
open science

Softbots Supporting the Operator 4.0 at Smart Factory Environments

Ricardo J. Rabelo, David Romero, Saulo Popov Zambiasi

► **To cite this version:**

Ricardo J. Rabelo, David Romero, Saulo Popov Zambiasi. Softbots Supporting the Operator 4.0 at Smart Factory Environments. IFIP International Conference on Advances in Production Management Systems (APMS), Aug 2018, Seoul, South Korea. pp.456-464, 10.1007/978-3-319-99707-0_57. hal-02177873

HAL Id: hal-02177873

<https://inria.hal.science/hal-02177873>

Submitted on 9 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Softbots Supporting the Operator 4.0 at Smart Factory Environments

Ricardo J. Rabelo¹, David Romero², Saulo Popov Zambiasi³

¹ UFSC - Federal University of Santa Catarina, Florianopolis, Brazil
ricardo.rabelo@ufsc.br

² Tecnológico de Monterrey, Mexico
david.romero.diaz@gmail.com

³ UNISUL - University of Southern Santa Catarina, Florianopolis, Brazil
saulopz@gmail.com

Abstract. This paper investigates how software robots, also known as softbots, can support the Operator 4.0 in smart factory environments, helping in the interfacing between smart machines and computer information systems with the aims of supporting the Operator 4.0 in different tasks at the shop floor. The work uses a reference framework called ARISA, which allows the derivation of softbots for given domains. An experimental setup and its results are presented in a testing scenario of a softbot to support the Operator 4.0 concept.

Keywords: Industry, 4.0, Operator 4.0, Human Cyber-Physical Systems, Human-Machine Interfaces, Smart Factories, Social Internet of Things, Services and People, Software Robots, Softbots, Socially Sustainable Manufacturing.

1 Introduction

Industry 4.0, or *cyber-physical production systems*, is a new concept being gradually adopted by manufacturing enterprises in order to increase their general efficiency and sustainability while coping with the need of highly customized and shorter lifecycles products and emerging product-service systems [1]. Benefiting from the advances on industrial automation, information and communication technologies (ICTs) and control and management models – shopfloor systems and equipment have turned into much more active entities within the wider, intensively collaborative and smarter production environment that characterizes the Industry 4.0 scenario [2].

A number of core-systems' design principles have been considered as a must to be supported by manufacturing enterprises when adopting Industry 4.0 architectures, platforms and technologies [1-3]: interoperability, modularity, virtualization, real-time information, service-orientation, and decentralization/autonomy.

Although much emphasis has been put on the “automation and control” part of the digitalization process of production systems, including on the so-called *cyber-physical systems*, many research works in literature have underestimate the impact of the Industry 4.0 on the workers and, at the same, on the new systems' requirements needed to support the new types of human-machine interactions arisen with such new systems generation (i.e. human cyber-physical systems) [4-6].

The concept of *Operator 4.0* [4] [5] has emerged to embrace this issue, promoting a balanced and/or symbiotic interaction between humans and machines. It aims to support a socially sustainable manufacturing workforce environment in the factories of the future, where “smart and skilled operators should perform not only ‘cooperative work’ with robots, but also ‘work aided’ by machines as and if needed, by means of

human cyber-physical systems, advanced human-machine interaction technologies and adaptive automation towards ‘human-automation symbiosis work systems’ ” [4].

In spite of the conceptual advances on the *Operator 4.0* area (e.g. [4-6]) and based on a literature review about it, it has been realized that larger efforts are necessary *to implement the Operator 4.0 vision*. This is the underlying motivation of this paper: How to ‘translate’ this concept into practice?

A number of approaches can be used for that. One of them is via *software robots*. A software robot, also known as *softbot*, can be generally defined as a virtual system deployed in a given computing environment that automates and helps humans in the execution of some tasks with variable levels of intelligence and autonomy [7].

In this direction, this paper presents a proof-of-concept and qualitative work that aims at investigating at which extent ‘softbots’ can support the implementation of the *Operator 4.0 concept*, particularly the *Smarter Operator 4.0 type* [5]. This is a type which is helped by softbots as *Intelligent Personal Assistants (IPAs)* to interface with smart machines and robots, computers, databases and other information systems so as to aid the operator in the execution of different tasks in a human-like interaction. This type of Operator 4.0 is grounded on the essentials of Industry 4.0 reference architectures design principles (viz. *Reference Architecture Model for Industrie 4.0 - RAMI 4.0* (Germany) and *Industrial Internet Reference Architecture - IIRA* (USA)).

A *reference framework* for deriving instances of ‘softbots’ was conceived in a previous work [8] and it was used to create a prototype of a “Smarter Operator 4.0”.

This paper is organized as follows. Section 1 has presented the problem and the paper’s goal. Section 2 gives a brief review about softbots and related works. Section 3 describes a reference framework and the model used to derive softbots. Section 4 presents the developed prototype and some current results. Section 5 summarizes the conclusions so far got as well as the main next steps of this research.

2 Softbots and Related Works

The research on *softbots* is not really new*, but their “popular usage” in industry is. Literature [9-14] shows their increasing use by large software-houses (e.g. Microsoft, Apple and Google) mainly from the last decade on. *Softbots* are becoming more and more used by general companies, for example in the traveling, health, banking and government sectors, where *softbots* interacts with users to solve problems, to clarify issues, etc. – typically using ‘chatbots’ or ‘intelligent personal assistants’ (IPAs).

A number of relevant *softbot implementations* (AI-based chatbots or IPAs) have been developed since then, such as ‘Cortana’ [9], ‘Sandy’ [10], ‘Siri’ [11], ‘PAL’ [12], ‘Narval’ [13], and ‘Alexa’ [14]. Their applications cover several different specific goals, like recalling users about different daily appointments and tasks; performing repetitive business activities autonomously on behalf of the user; searching for solutions of problems through the Internet considering users’ profile, problem’s context and historical data; proposing solutions for unexpected situations via machine learning techniques; among others.

Actually, two basic types of *chatbots* can be found in literature [15]: (a) *Rule-based chatbots*, programmed to recognize certain terms and patterns from which they

* First chatbot, named: Eliza (1996) – by Massachusetts Institute of Technology (MIT).

can respond with pre-set answers (e.g. MS-Office Assistant *Clippy*); and (b) *Artificial Intelligence (AI)-based chatbots*, which are considered as ‘artificial brains’ using sophisticated cognitive and natural language processing capabilities for understanding the context, intent and emotion(s) of users’ requests, and that evolves as they interact and learn from conversations with users (e.g. Microsoft AI *Cortana*).

A *softbot* can implement many different things within several domains. Romero et al. [5] have identified eight categories of possible applications where human operators might be aided by *intelligent software* in Industry 4.0 [4] [6]. Its interaction with humans can be provided by different means, like web-browsers and desktop computers, mobile devices, holography, augmented reality, natural language, haptics, etc.

BCG [16] has recognized the importance of *software assistance* in the Industry 4.0 and outlined ten so-called ‘use cases’ scenarios to be supported: (a) *Big-data-driven quality control*, where ‘softbots’ can make use of data analytic techniques to support quality engineers in analysing real-time and historical quality-control data, identifying quality issues and their causes and pinpointing ways to minimize product failures and waste; (b) *Robot-assisted production* and (c) *Self-driving logistics vehicles*, where ‘softbots’ can oversee all the industrial entities operations (viz. smart machine tools, robots, co-bots, belt-conveyors, AGVs (i.e. automated guided vehicles) and human operations) on the shopfloor in order to guarantee humans and machines safety in advanced human-machine interactions (HMIs) and the overall manufacturing system’s productivity; (d) *Production line simulation*, where ‘softbots’ can provide proactively production planners with insights on how to optimize and support all operational demands; (e) *Smart supply network*, where ‘softbots’ can provide supply network managers with real-time monitoring (e.g. track & trace) and simulation (e.g. what-if scenarios) capabilities to enable companies to react or even anticipate disruptions in the supply network, and therefore, adjust the network in advance or in real-time as conditions change; (f) *Predictive maintenance*, where ‘softbots’ can support service technicians with automated remote alerts, delivered in their mobile devices, based on real-time monitoring of smart equipment and with intelligent embedded prognostics algorithms as troubleshooting assisting means; (g) *Machine-as-a-service* and (h) *Self-organizing production*, where ‘softbots’ can support production managers with information about the location, condition, availability, etc. of smart equipment for assisted production management (i.e. coordination and optimization of the utilization of each production asset), (i) *Additive manufacturing of complex parts*, where ‘softbots’ can help industrial designers to determine the “printability” of given 3D-objects acting as “printability checkers” using machine learning techniques, and (j) *Augmented work, maintenance, and service*, where ‘softbots’ can help to dispatch (i.e. find, sort, filter and deliver) the right information to the operator in abundant information environments (the smart factory) as part of digital assistance systems.

Schwartz et al. [17] proposed a concept of “hybrid teams” to face the increasing need for higher-level collaboration between humans, smart industrial equipment and software so as to better underpinning Industry 4.0 requirements (see also [6]).

Older but very powerful approaches, as *multi-agent systems* and *holonic systems*, seem to be ‘rediscovered’ after their boom in the end of 90s in a time where Internet-based ITs were just flourishing [18]. Their intrinsic properties are pretty much in line with the Industry 4.0 architecture design principles, especially in what the decisions’ decentralization and the autonomy of equipment and industrial entities is concerned.

Several works [e.g. 18-20] were developed based on those approaches, where agents could virtualize and represent industrial entities from any type in such way they could be able to reason about the current shopfloor status, to autonomously provide real-time information to other systems, to dynamically establish opportunistic consortia to collaborate, and to solve problems via e.g. negotiation strategies.

Despite the relevance of all those mentioned works, the ones presenting real implementation of *softbots (or equivalent)* were not properly devised to cope with *Industry 4.0* and *Operator 4.0 scenarios*. On the other hand, the works presenting implementations related to *Industry 4.0* were neither devised to cope with *Operator 4.0 scenarios* nor to support *softbots* to help humans in their activities.

3 Softbots Reference Framework – Supporting the Operator 4.0

A *reference framework* called ‘ARISA’, presented in [8], has been used to develop the intended proof-of-concept *Smarter Operator 4.0 type* as a *softbot implementation*. This framework was chosen due to its main intrinsic properties, which offers some support to implement (at different levels of depth) the core design principles of *Industry 4.0 architectures* (Table 1).

Table 1. ARISA: Supporting Core-Design Principles of Industry 4.0 Architectures

Interoperability	<ul style="list-style-type: none"> It is strongly based on open-IT standards and integration patterns, facilitating interoperability with human end-users, other softbots, with other shopfloor entities and systems (including legacy systems), and with other enterprise systems (e.g. ERPs, Amazon Cloud and Facebook 's APIs).
Modularity	<ul style="list-style-type: none"> The used services can be implemented from scratch, but can also use services provided by open digital software services ecosystems or IT business partners, by means of static or dynamic services compositions. New services can be added and other ones can be deleted from the softbot without damaging its internal architecture. The softbot's internal architecture can be designed following the 3-Tiers or MVC models, and its user interface can be showed through different devices (e.g. mobile devices). Yet, the different model's layers can be deployed at different servers.
Virtualization	<ul style="list-style-type: none"> It allows the derivation of instances-of 'softbots' for different domain applications.
Real-time Information	<ul style="list-style-type: none"> A softbot can perform several actions simultaneously and asynchronously.
Service-orientation	<ul style="list-style-type: none"> Its instances are natively designed under the Service-Oriented Architecture (SOA) model, meaning that a given softbot is internally composed of a set of built-in (distributed) software services, such as communication ways (as Twitter, webmail, text entry and voice); and standards reports, for e.g. communication auditing. It is also open to implement as many services as needed for particular applications.
Decentralization/ Autonomy	<ul style="list-style-type: none"> The softbot's behaviour (including the autonomy and intelligence levels) can be variable and implemented for different scenarios.

The *ARISA Reference Framework* is actually the implementation of the *ARISA Reference Model* in concrete software artefacts so as to support particular derivations, as the envisaged IPA (see Figure 1).

In very general terms and in a high-level view, the model has as elements: the *User* interacts with his/her IPA using proper interfaces – *User Applications* (e.g. Gmail, Twitter, text-consoles, etc.); a *Toolbox* (general programming & configuration tools to derive particular IPAs); and *Legacy Applications* (enterprise's applications that can be involved in the business processes the derived IPA should interact with).

The *Personal Assistant Manager* represents the IPA's core, being responsible to define, configure and manage the right access and execution of all internal IPA's entities. The *Services Federation* involves all registered software services that can be involved in the IPA execution regarding its goals and related business processes. The *Interoperability Services* refer to all software required to support the interoperability between the systems and general computing artefacts used once an IPA is derived. The *Toolbox* comprises all the supporting tools, execution engines and computing environments involved in the integration and deployment of the IPA's elements.

ARISA is organized in the *SOA (Service Oriented Architecture)* style, meaning that all the IPA's elements are modelled and implemented as software services. By default, the IPA is designed to work under the 3-Tiers model, being the many of the involved services physically separated into presentation, process and data layers.

A number of activities have to be carried out in order to derive an IPA. To be emphasized the activities related to define the IPA's business processes and activities to be executed (i.e. *what*) of a given problem area; the IPA's behaviour (i.e. *how* and *when* to do the *what*); the IPA's resources (i.e. *which* internal or external software services will execute the *what*, where services can be coded in different languages, like in *PHP*); and the IPA's execution places (i.e. *where*), when it is necessary to endow it with the ability to 'travel' through the network to execute some of their tasks in another server or equipment, as a mobile agent [21]).

ARISA has a specific programming shell to do that (one of the Toolbox's tools). It can be seen as a lower-level *BPM (Business Process Management)* modelling in the form of *if-then-else-end* statements to express the business process' logic as well as the respective internal and external services to be invoked (see Figure 2). A *BPEL (Business Process Execution Language)* file is then generated, which runs quite well regarding that only *web-services* and *SOAP* protocol are supported in this current implementation version.

4 Experimental Setup and Results

Considering that this is a proof-of-concept work and regarding the *Operator 4.0 typology*, a small scenario to support the *Smarter Operator 4.0 type* was built up.

ARISA is actually an academic prototype and it is not very easy to derive instances of softbots and to configure it, which requires an expert on IT.

The devised testing scenario comprises a lathe machine and its CNC controller, its respective *Operator 4.0*, and a buffer that feeds the smart machine considering a given batch-size of raw parts to be machined.

The business process (BP) was designed to monitor the number of parts in such way the machining can keep operating only if there is a minimum number of parts (10) of type '101' in the buffer. There is a sensor in the buffer which counts this number. This information is sent out (via a emulated *Profinet Industrial Network*) to the CNC controller's wrapper (compliant to the ISO 9506 MMS/VMD) when the buffer reaches the minimum level so that a '*purchasing order*' (business process of id '80') can be triggered and performed by the IPA close to the part's supplier (called hypothetically as '*John Doe Ltda*'). This transaction will cost \$25.00.

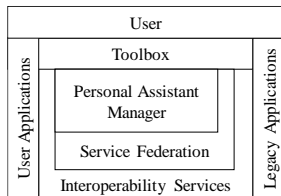


Fig. 1. ARISA Reference Model



Fig. 2. Interface for the IPA Management

However, this supplier does not have the required total amount of parts and the IPA autonomously decide to look for another supplier. It finds the supplier 'Sora konpyuuta' company in the enterprise's supplier catalogue, but the cost of this transaction (for the 17 required additional parts) is \$28.00. The *Operator 4.0* - 'Mr. Saulo' is notified that there are 17 parts needed to be purchased, via an order '81', to complete the BP production.

Three basic use cases were supported: (i) The IPA manages the whole mentioned process scenario and keep the *Operator 4.0* informed about the purchasing success via the IPA's graphical interface, using the *Gtalk* tool (Fig. 3); (ii) the *Operator 4.0* interacts with the IPA via text or voice in his/her mobile phone asking about how many orders are still pending in that machine, using the *Twitter* tool. In this case, the IPA answers that the order '84' is still opened, and it is related to '14' units of product '101' with a cost of \$28.00 (Fig. 4); and (iii) the IPA (the ARISA Personal Assistant) publishes as a report a summary of its daily activities, using a *blog* tool (see Fig. 5).

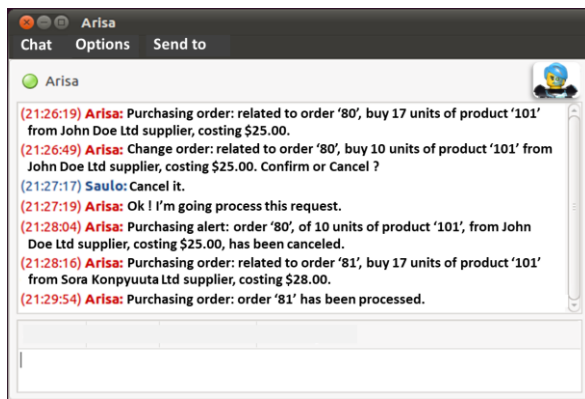


Fig. 3. IPA's Graphical Interface



Fig. 4. Operator 4.0 Mobile Phone

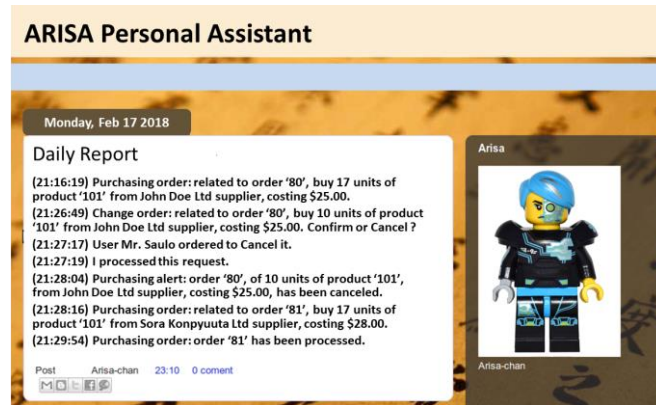


Fig. 5. IPA's Daily Report

5 Conclusions and Further Research

This paper has presented a proof-of-concept on how *softbots* could be used as a feasible approach to implement the *Smarter Operator 4.0 type* [6]. The implemented *softbot* was derived from a reference framework called 'ARISA', which natively supports the Industry 4.0 architectures design principles.

The development of 'human-automation symbiosis' offers advantages for the sustainability of the manufacturing workforce in Industry 4.0, improving operational excellence, inclusiveness, satisfaction and motivation, safety and continuous learning.

A use case scenario was implemented in a controlled and emulated shopfloor environment. Based on this, it can be said that *softbots* can fulfil several of the "Smarter Operator 4.0 type" interfacing requirements with other industrial entities.

The *softbot prototype* could perform the actions as planned and hence could assist the human operator in some tasks, both in terms of interacting with him/her when needed and by automating tasks execution on behalf of him/her. These were the general performance indicators used to evaluate this current version of the prototype.

Important to mention that part of this successful results was due to the ARISA framework, which uses many integration patterns as well as supports a high-level of flexibility and configurability when deriving instances-of IPAs.

There are some technological limitations in the developed prototype. For example, there is no support for security, semantic interoperability, advanced voice processing, robust natural language recognition, advanced usability techniques, and adaptive *softbot's* behaviour. On the other hand, thanks to the intrinsic SOA-based architecture of derived *softbots*, and to the fact that the internal architecture is open, standard-based and scalable, those limitations can be mitigated using specific services for that, which in turn can be invoked from external/proper services providers.

The implemented scenario was relatively simple, covering a business process related to the general management of an equipment, including both human interactions and automatic actions by the IPA. In a real-case scenario the modelling of all the necessary business processes and interaction with human can be very hard to conceive, model, implement and maintain. Besides that, business processes are very different to each other, meaning that critical actions may have to be supervised or authorized by humans before being taking. All this requires deep training of workers.

This is an ongoing work. Next main steps comprise improvements on the voice and natural language recognitions, evaluation of softbots in other types of the *Operator 4.0 typology*, and deeper analysis of integration approaches between the softbot and real smart industrial equipment's wrappers and controllers.

References

1. Hermann, M., Pentek, T., Otto, B.: Design Principles for Industry 4.0 Scenarios. Proc. 49th IEEE Hawaii International Conference on System Sciences, pp. 3928-3937 (2016)
2. Lee, J., Bagheri, B., Kao, H.A.: Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial informatics. IEEE, Industrial Informatics (2014)
3. Sameer, M., Muztoba, K., Romero, D., Wuest, T.: Smart Manufacturing: Characteristics, Technologies and Enabling Factors. Journal of Engineering Manufacture (2017)
4. Romero, D.; Bernus, P.; Noran, O.; Stahre, J. and Fast-Berglund, Å.: The Operator 4.0: Human Cyber-Physical Systems & Adaptive Automation towards Human-Automation Symbiosis Work Systems. Proc. APMS 2016 Conference, pp. 677-686 (2016)
5. Romero, D., Stahre, J., Wuest, T., Noran, O., Bernus, P., Fast-Berglund, Å., Gorecky, D.: Towards an Operator 4.0 Typology: A Human-Centric Perspective on the Fourth Industrial Revolution Technologies. Proc. International Conference on Computers & Industrial Engineering, pp. 1-11 (2016)
6. Romero, D., Wuest, T., Stahre, J., Gorecky, D.: Social Factory Architecture: Social Networking Services and Production Scenarios through the Social Internet of Things, Services and People for Operator 4.0. Proc. APMS 2017 Conference, pp. 265-273 (2017)
7. Kim, J.H.: Ubiquitous Robot. In: Reusch B. (Eds.) Computational Intelligence, Theory and Applications. Advances in Soft Computing, Vol 33, Springer, (2005)
8. Zambiasi, S.P., Rabelo, R.J.: A Proposal for Reference Architecture for Personal Assistant Software based on SOA. IEEE Latin America Transactions, 10(1):1227-1234 (2012)
9. Cortana, <https://developer.microsoft.com/en-us/cortana>
10. Sandy, <https://boingboing.net/2007/11/14/i-want-sandy-perfect.html>
11. Siri, <https://www.apple.com/ios/siri/?cid=oas-us-domains-siri.com>
12. PAL, DARPA PAL Project, <http://pal.sri.com/>
13. Narval, C.N.: The Intelligent Personal Assistant or How the French Linux Gazette is Built, <https://linuxgazette.net/issue59/chauvat.html>
14. Alexa, <https://developer.amazon.com/alexa>
15. Monchalin, E.: <https://atos.net/en/blog/an-introduction-to-chatbots> (2017)
16. BCG Boston Consulting Group: <https://www.bcg.com/pt-br/publications/2015/technology-business-transformation-engineered-products-infrastructure-man-machine-industry-4.aspx>
17. Schwartz, T., Zinnikus, I., Krieger, H.U., Bürckert, C.: Hybrid Teams: Flexible Collaboration between Humans, Robots & Virtual Agents. Proc. MATES'2016 – The 14th German Multiagent System Technologies Conference, pp. 131-146 (2016)
18. Rabelo, R.J., Camarinha-Matos, L.M.: Negotiation in Multiagent Dynamic Scheduling. Robotics and Computer-Integrated Manufacturing Journal, 11(4):303-310 (1994)
19. Barata, J., Camarinha-Matos, L.M.: Coalitions of Manufacturing Components for Shop Floor Agility. Int. Journal of Networking and Virtual Organisations, 2(1):50-77 (2003)
20. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L.: Reference Architecture for Holonic Manufacturing Systems. Computers in Industry, 37(3):255-274 (1998)
21. Wangham, M. S., Fraga, J. S., Rabelo, R.J., Lung, L.C.: Secure Mobile Agent System and its Application in the Trust Building Process of Virtual Enterprises. In Multiagent and Grid Systems Journal, 1(1):147-168 (2005)