



INDUSTRIAL AGENTS

Emerging Applications of
Software Agents in Industry

Edited by Paulo Leitão and Stamatis Karnouskos

Multi-agent System for Integrating Quality and Process Control in a Home Appliance Production Line

Paulo Leitão^{1,2}, Nelson Rodrigues^{1,2}, Claudio Turrin³, Arnaldo Pagani³

¹ Polytechnic Institute of Bragança, Portugal {nrodrigues, pleitao}@ipb.pt

² LIACC - Artificial Intelligence and Computer Science Laboratory, Portugal

³ Whirlpool Europe, Italy {Claudio_Turrin, Arnaldo_Pagani}@whirlpool.com}

1.1 Introduction / Motivation	1
1.2 Application Overview	2
1.2.1 Problem description	2
1.2.2 Agent-based architecture	3
1.2.3 Cooperation patterns and ontology.....	6
1.3 Application Details	8
1.3.1 Implementation of the agent-based solution	8
1.3.2 Installation in the factory plant.....	10
1.4 Benefits and Assessment	12
1.4.1 Qualitative properties	12
1.4.2 Quantitative impact.....	13
1.5 Discussion	14
1.6 Conclusions	15
Acknowledgments	15
References	16

1.1 Introduction / Motivation

A current trend in manufacturing is the deployment of modular, distributed and intelligent control systems that introduce adaptation facing unexpected deviations and failures, namely in terms of production conditions and product demand fluctuation. The integration of quality and process control allows the implementation of dynamic self-adaptation procedures and feedback control loops to address a large variety of disturbances and changes in process parameters and variables, aiming to improve the production efficiency and the product quality.

Multi-agent systems (MAS) technology (Wooldridge 2002)(Leitão et al. 2013) is suitable to face this challenge, offering an alternative way to design these adaptive systems, based on the decentralization of functions over distributed autonomous and cooperative agents, providing modularity, flexibility, adaptation and robustness. In spite of the potential benefits of the MAS technology, the number of deployed agent-based solutions in industrial environments, reported in the literature, are few, as illustrated in (Leitão et al. 2013) [colocar aqui referencia ao Pechoucek & Marik].

This chapter describes the development, installation and operation of a multi-agent system, designated as GRACE, integrating quality and process control to operate in a real home appliance production line, producing laundry washing

machines, owned by Whirlpool and located in Naples, Italy. The use of the MAS technology acts as the intelligent and distributed infra-structure to support the implementation of real-time monitoring and feedback control loops that apply dynamic self-adaptation and optimization mechanisms to adjust the process and product variables. The agent-based solution was developed using the JADE (Java Agent DEvelopment Framework) framework and successfully installed in the industrial factory plant, contributing for demonstrating the effective applicability and benefits of the MAS technology, namely in terms of production efficiency and product quality.

1.2 Application Overview

1.2.1 Problem description

The problem addressed in this work considers a washing machine production line, composed of several workstations arranged in a sequential order, and linked together by conveyors. Each individual workstation performs one operation in the product being produced, which can be from different types: processing (e.g. bearing insertion, seal insertion or pulley screwing), quality control (e.g. visual inspection, vibration analysis or functional tests inspection) or manual operation (e.g. cable and electronics assembly). The quality control stations are distributed along the production line to verify the product quality conformity. As an example, the functional tests station comprises 6 boxes to perform a fixed plan of functional tests to all washing machines produced in the production line, which lasts 6 minutes for each one. A simplified vision of the production line is illustrated in Fig. 1.

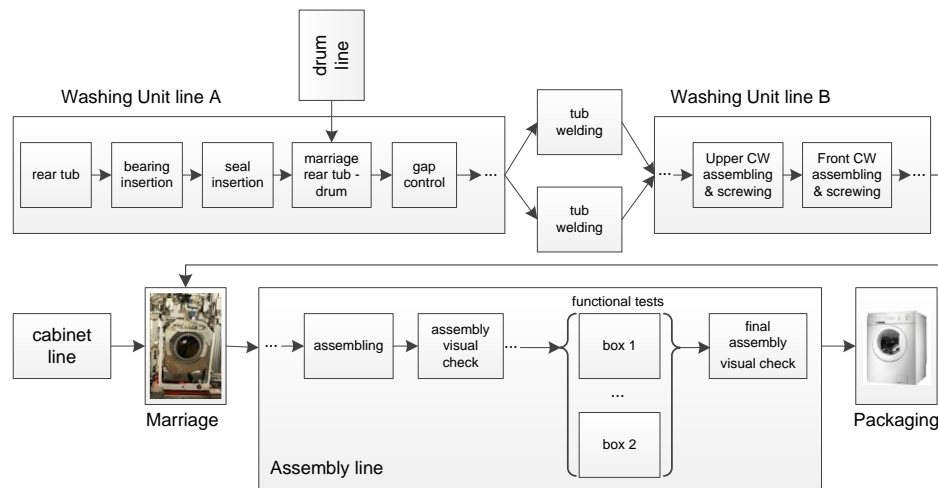


Fig. 1. Layout of the production line.

The objective is the integration of the process and quality control levels, creating feedback control loops to allow the adaptation of production parameters, improving the product quality and the process performance, e.g. reducing the production time, correcting earlier the deviations or quality

problems, skipping unnecessary tests along the line and customizing the final product.

An important assumption in this work is to maintain the existing low-level control in the shop floor, which already uses the state-of-the-art industrial control based on Programmable Logic Controllers (PLCs) running IEC 61131-3 control programs, and introduce the MAS solution at a higher control level to provide intelligence and adaptation.

1.2.2 Agent-based architecture

The designed agent-based system, illustrated in Fig. 2, comprises a society of autonomous and cooperative agents representing the manufacturing components disposed along the production line. These agents may represent a physical resource, e.g., a processing or a quality control station, or a logic activity, e.g. an order.

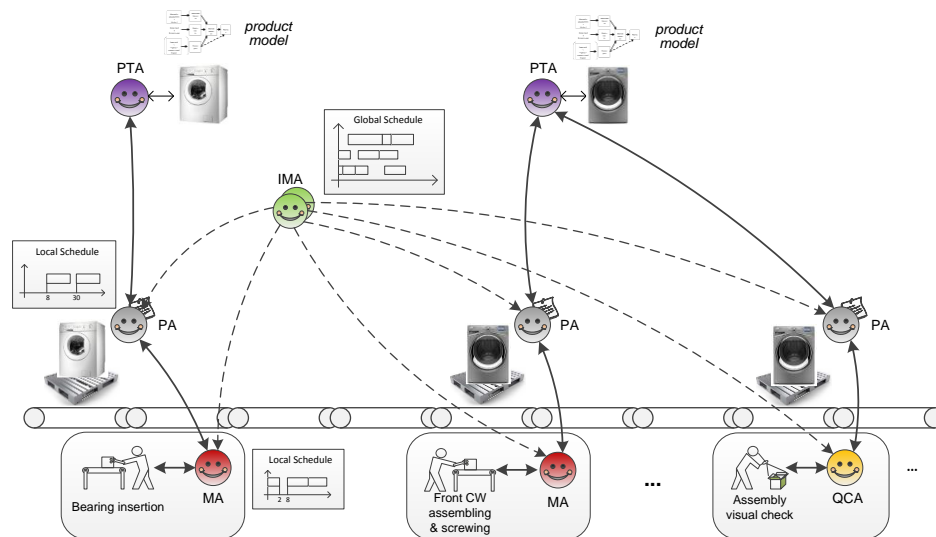


Fig. 2. Multi-agent System Architecture for the Production Line.

Several types of agents were identified, according to the process to control and to their specialization, namely (Leitão & Rodrigues 2011):

- *Product Type Agents (PTA)*: representing the catalogue of products that can be produced in the production line (one agent for each type of product). These agents contain the process and product knowledge required to produce the product.
- *Product Agents (PA)*: handling the production of product instances being produced along the production line (one agent for each washing machine). They possess the process plan to produce the product and interact with the agents responsible for the process and quality control.
- *Resource Agents (RA)*: representing the resources disposed along the production line, namely processing stations, quality control stations and operators. Several specializations were considered according to the type of operations performed, namely Machine Agents (MA), associated to

processing machines, such as welding robots or screwing stations, and Quality Control Agents (QCA), associated to quality control stations.

- *Independent Meta Agents (IMA)*: acting at strategic level, taking advantage of their global perspective, to provide global optimization to the production system. In opposite to PA and RA agents, which are placed at the operational execution level and are mandatory, the IMA agents are positioned in a higher strategic level and are not mandatory (i.e. the system can continue working without them, however losing some optimization).

The GRACE agent-based solution follows the ADACOR principles (Leitão & Restivo 2006), namely using a Petri nets methodology to specify the system behaviour. In this way, the behaviour of each individual agent is modelled by using the Petri nets formalism (Murata 1989), taking advantage of the well-founded mathematical theory that allows to design the control system behaviour, but also to validate the system specifications, supporting the design-implementation path. In this work, the modelling phase is illustrated by modelling the behaviour of the RA agent, which Petri nets behavioural model is illustrated in Fig. 3.

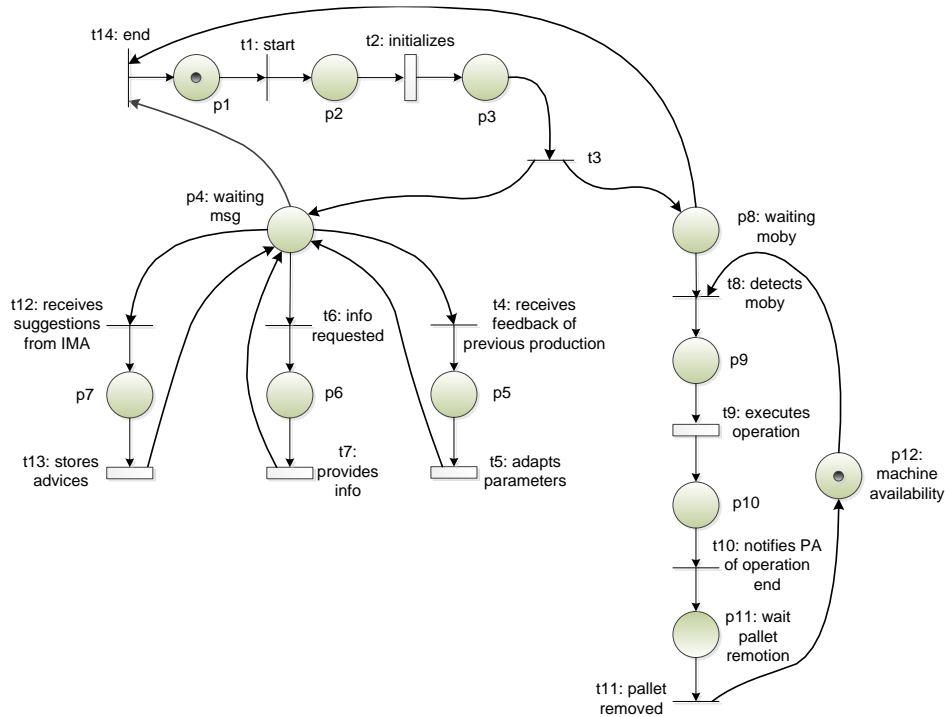


Fig. 3. Behaviour of the RA agent using the Petri nets formalism.

Briefly, after the initialization phase, the agent enters into two sub-behaviours running in parallel, related to managing the operation execution (represented by the place p_8) and waiting for monitoring requests or feedback suggestions (p_4). The operation execution is started when the station detects the arrival of the pallet, represented by the transition t_8 and using a RFiD (Radio-Frequency

Identification) reader. This activity, i.e. the timed transition t_9 , can be expanded into a more refined Petri nets model containing the details about the executed functions, as illustrated in Fig. 4. During the execution of the operation activity, the agent may adjust the operation parameters, by applying self-adaptation procedures according to its local knowledge, and triggers the execution of the operation by sending a command to the resource (i.e. the physical equipment or operator). At the end of the processing or testing operation, the gathered data is analysed, and the results are sent to the PA and IMA agents, to be used in future adaptation procedures.

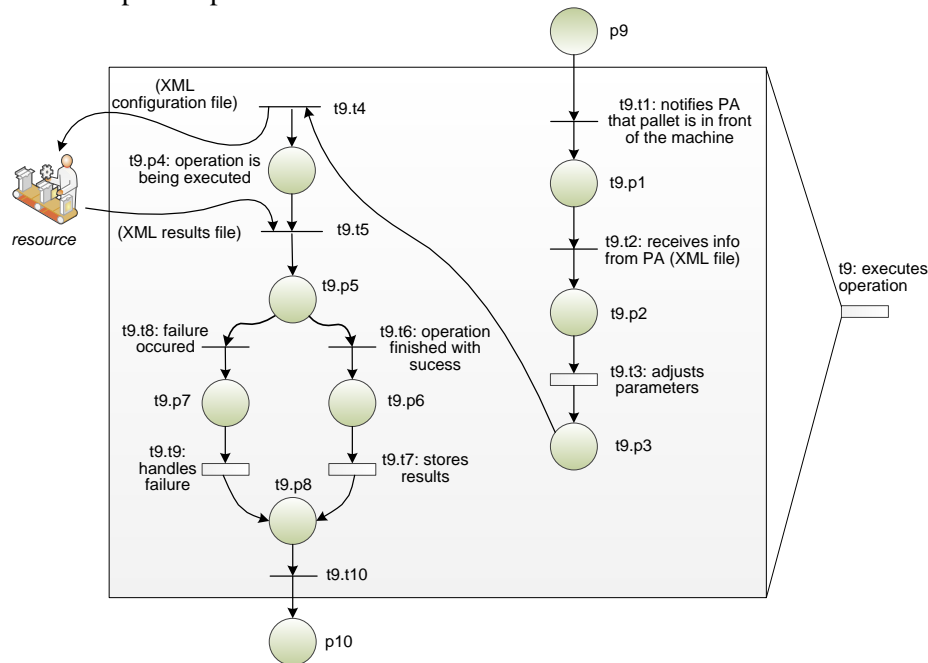


Fig. 4. Sub-Petri nets model for the activity "Executes operation".

In parallel to this behaviour, the agent may receive information of the execution of the previous operations, sent by QCAs after executing inspection tests, which will support the improvement of its performance in the future (e.g., by using self-learning procedures in the activity "adapts parameters" represented by the timed transition t_5). Also, the suggestions sent by IMA agents are stored in the local database for posterior usage.

The Petri net model can be validated, using a qualitative (structural and behavioural) and quantitative (performance/simulation) analysis, not only to prove that the model is free of errors, but also to test if it performs the desired specifications. As example, the results from the analysis of the behavioural properties of the RA Petri nets model are illustrated in Fig. 5. From this analysis it is possible to verify that the model is bounded, safe (e.g. meaning that it is only possible to execute only one operation at each time), reversible and is absent of deadlocks.

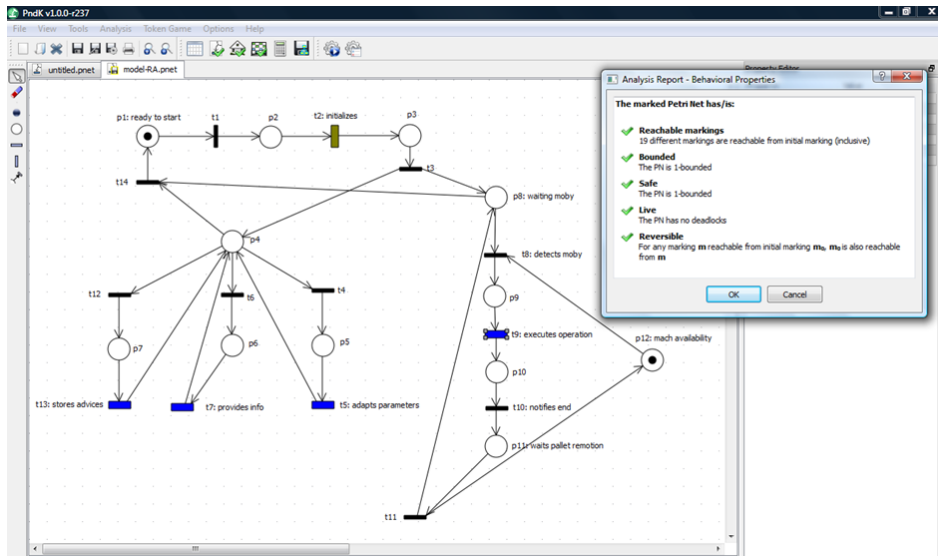


Fig. 5. Behavioural analysis of the RA Petri nets model.

A similar procedure was applied to formally specify the dynamic behaviour of the others agents belonging to the system.

1.2.3 Cooperation patterns and ontology

The overall control system for the entire production line emerges from the cooperation among the distributed, autonomous agents, coordinating their actions according to the agents' goals. Of particular importance are the cooperation patterns aiming the integration of process and quality control, closing the control loops and implementing self-adaptation procedures. Examples are the dynamic adaptation of the functional tests plan, the customization of the on-board electronic controller and the generation of warnings in any point of the production line in case the desired quality is not possible to be achieved anymore.

Fig. 6 illustrates one example of these interaction diagrams, showing the interaction among the agents during the process of adaptation of the functional tests plan. Note that near the end of the production line, a quality control station is devoted to perform a set of functional tests to the produced washing machines. The knowledge of the historical production data allows to customize the plan of tests by removing unnecessary tests, adjusting others or customizing the messages to the operators.

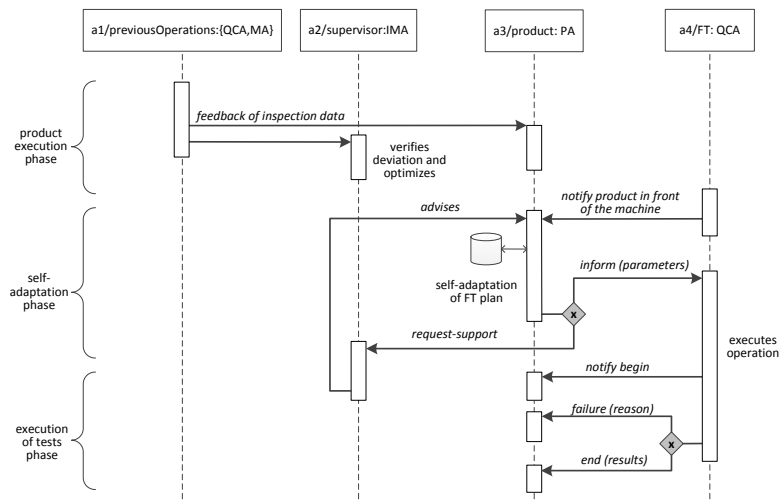


Fig. 6. Cooperation pattern for the adaptation of the functional tests plan.

Along the production line, the PA agent is receiving feedback information related to the results of the execution of the processing and inspection operations, respectively from MA and QCA agents. When the PA arrives to the functional tests station, it executes a self-adaptation procedure to customize the plan of tests for this particular washing machine, correlating the historical production data of the washing machine. In parallel, the PA can request support from IMA agents, which based on their wider perspective can provide advises for the execution of the self-adaptation procedure.

The interaction among individual agents is crucial in MAS applications, requiring that the agents can understand themselves to share knowledge, using a proper agent communication language and a proper knowledge representation. Of special importance is the establishment of a common understanding among the agents, since the exchange of shared knowledge becomes difficult if each agent has its own knowledge structure representation. The solution is to use ontologies [colocar aqui link para a referencia para Gruber] to define a common understanding among distributed entities, defining the vocabulary and the semantics of the knowledge used in the communication between distributed agents.

In this work, an ontology was designed and implemented considering the particularities of the home appliance domain and the integration of process and quality control levels. This ontology formalizes the structure of the knowledge, namely the concepts, the predicates (relation between the concepts) and the terms (attributes of each concept), as illustrated in Fig. 7 (Leitao et al. 2012).

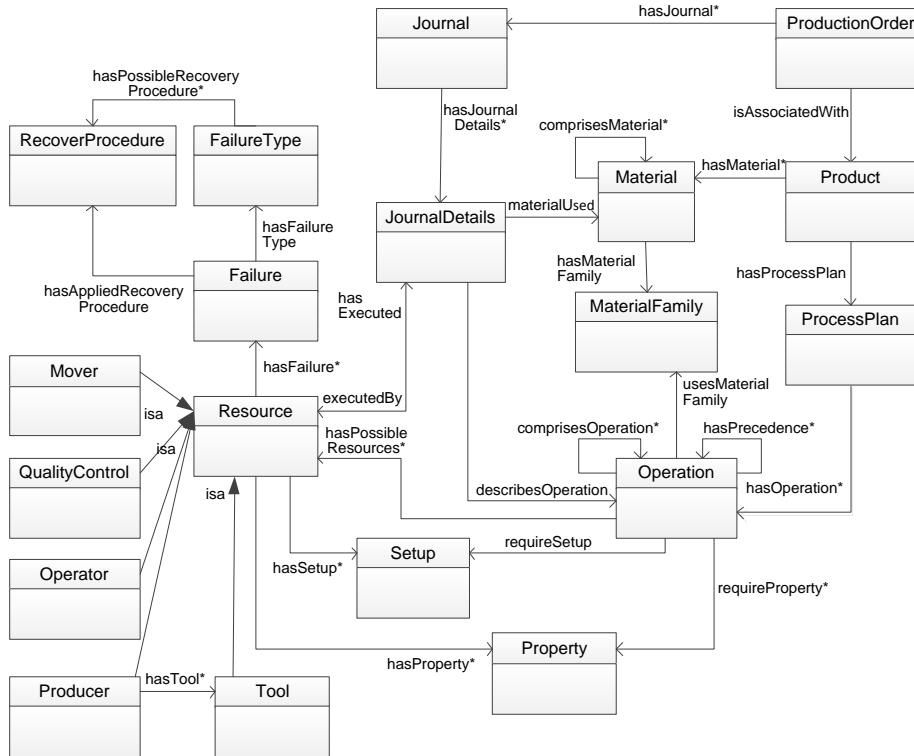


Fig. 7. GRACE ontology schema.

The GRACE ontology schema was edited and validated using the Protégé framework (Protégé 2014), which supports the Web Ontology Language (OWL) (W3C 2004) and has an easy connection with agent development frameworks, such as JADE.

At the end of the design phase, the MAS application is ready for being implemented and deployed using a proper MAS development framework.

1.3 Application Details

1.3.1 Implementation of the agent-based solution

The designed agent-based solution was implemented using the JADE framework (Bellifemine et al. 2007) and following the existing standards in the field, namely the FIPA (Foundation for Intelligent Physical Agents) specifications (FIPA 2014) (note that JADE is FIPA compliant). Each one of the developed agent types, namely PTA, PA, RA and IMA agents, is a simple Java class that extends the Agent class provided by the JADE framework. Each one inherits the basic functionalities from JADE, such as the white and yellow pages services, and the encoding and parsing of ACL messages, and extends them with features that represent the specific behaviour of the agent, as detailed in the previous section using the Petri nets models. Each agent is developed using multi-threaded programming, over the JADE's behaviour concept (Bellifemine et al. 2007), allowing the execution of several actions in parallel.

The communication between the distributed agents is asynchronous and done over Ethernet network using the TCP/IP protocol. The messages exchanged by the agents are encoded using the FIPA-ACL communication language, being their content formatted according to the FIPA-SL0 language. The meaning of the message content is standardized according to the designed GRACE ontology. The integration of the GRACE ontology, edited in Protégé, in the MAS solution was performed by using the OntologyBeanGenerator plug-in, which allows to automatically generate the Java classes from the Protégé tool, following the FIPA specifications. The main generated class represents the vocabulary and main ontological objects (i.e. concepts and predicates) defined in the ontology. The second group of generated Java classes specify the structure and semantics of each ontological object defined in the ontology. The integration of legacy systems, and particularly the interaction with physical devices hosted in the production line, assumes a critical role when deploying this kind of systems into industrial environments. As an example, QCA agents are associated to quality control functions, namely to the adaptation procedures allowing the improvement of the quality control station behaviour and consequently of the whole system behaviour. The interconnection between the QCA agent and the quality control station (QCS), illustrated in Fig. 8, comprises an intelligent part (the agent) and the physical part (the hardware device responsible for the inspection tasks), which in this work also embodies a measurement system developed in LabView™.

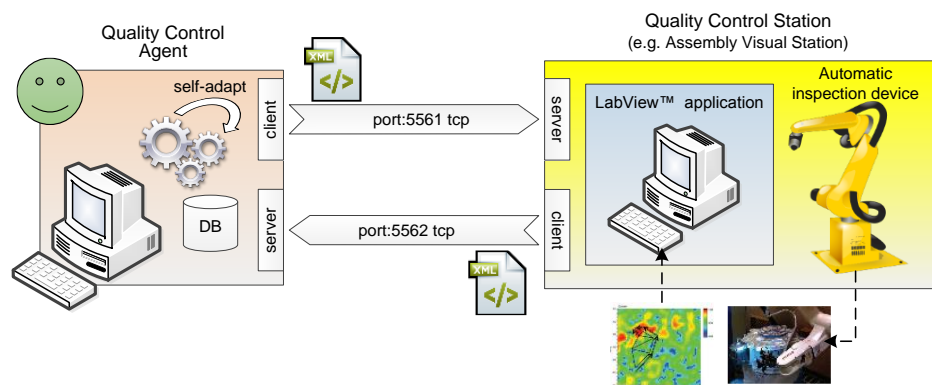


Fig. 8. Interconnection of QCA and QCS by the TCP/IP sockets to exchange information related to quality control tests.

Basically, the QCS application receives the data for the QCA, performs the inspection test and sends back the diagnosis result to the QCA. The approach used to perform this data exchange between QCA and QCS relies on opening two TCP/IP sockets: one for the communication from the QCA to the QCS and the other for the communication from the QCS to the QCA. The exchanged messages are described in XML (eXtensible Markup Language) to simplify the common understanding.

Several GUIs (Graphical User Interfaces) were developed using the Swing toolkit, allowing an easy interaction between the users and the agents.

1.3.2 Installation in the factory plant

The installation of the agent-based solution in the Naples factory required the distribution of the plethora of agents by 8 computers disposed along the production line, inter-connected by TCP/IP over an Ethernet network, as illustrated in Fig. 9.

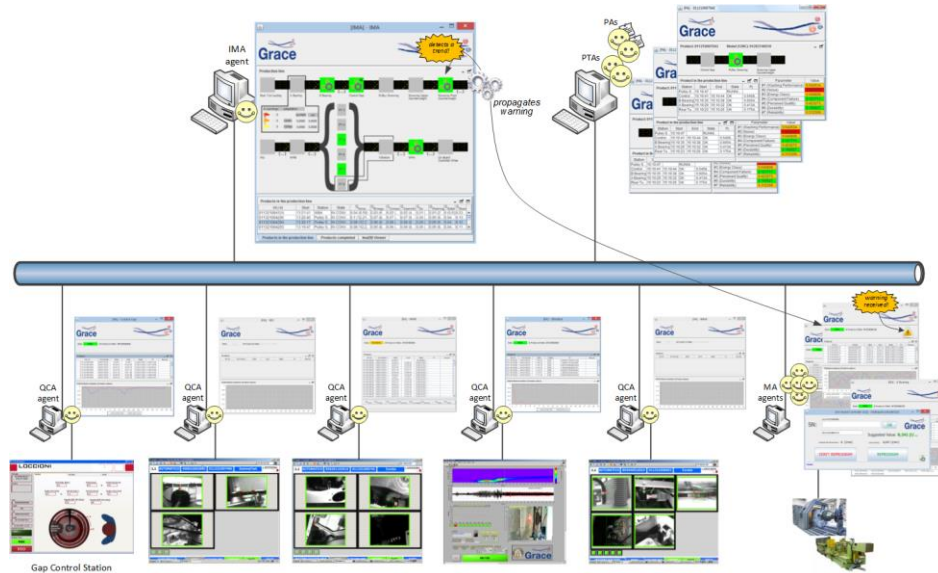


Fig. 9. Distribution of agents in the installed system at the Naples factory.

Each individual agent is customized according to its particularities, namely role, knowledge, skills and embedded algorithms, by interpreting a XML file during the start-up phase.

Particularly, the algorithms embedded in each agent aims to apply proper self-adaptation mechanisms, based on data mining and learning techniques, according to the scope of the agent. The following algorithms were implemented:

- The RA agents perform, in a continuous and individual manner, an analysis of the performance index of the processing/inspection station along the time, to detect trends or degradation in its behaviour. This allows to detect deviations in advance and to trigger the implementation of corrective measures, e.g. the automatic calibration of the tool wear or the light source, or to request external maintenance intervention, to mitigate the detected deviation.
- The PA agents perform, in a continuous and individual manner, an analysis of the evolution of the quality indexes of each washing machine along the production line, being detected deviations to the expected values specified for that model. This allows to detect, in real time and at any point of the production line, when the desired quality is not possible to be achieved anymore and consequently decide to stop the production of the washing machine to save time and money (since scraps can be reduced).

- Near the end of the production line, the PA agents apply an algorithm to adapt the plan of functional tests to be performed in every washing machine according to their production historical data, allowing to correlate the data properly. This permits to save inspection time and to execute more efficient inspections tests.
- The PA agents apply, at the end of the production line, a method to customize the parameters of the electronic controller installed in each washing machine to control its behaviour, based on the historical production data. This data can be related to the information of the previous operations of the washing machine and the information of other washing machines from the same model. For this purpose, the calculated customized parameters, e.g. the flow rate parameter, are written in the EEPROM (Electrically Erasable Programmable Read Only Memory) of the on-board controller.
- The PTA agents, based on the data supplied by PA agents at the end of the washing machines production, perform an adaptation of the process plan for each machine model, mainly adapting the parameters of individual operations.

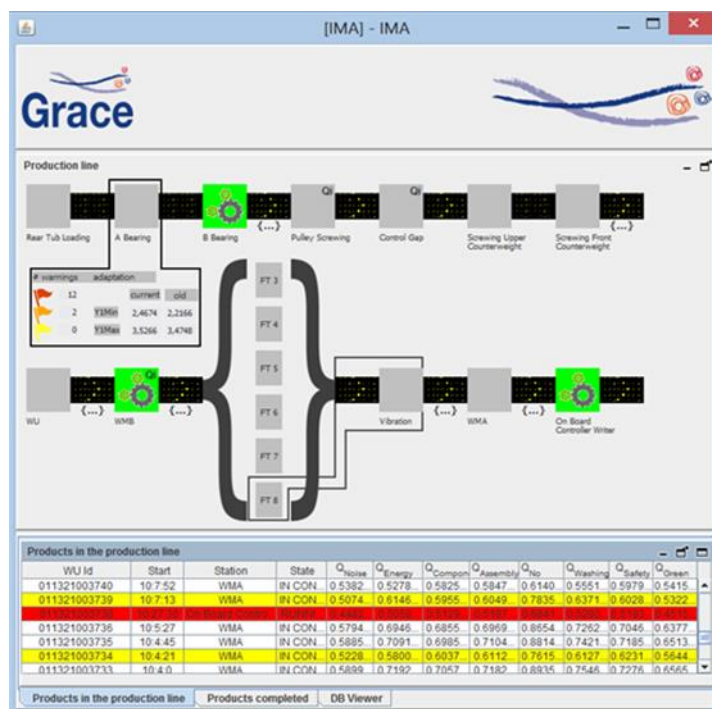


Fig. 10. Screenshot of the IMA agent, showing the current status of the production line and the information about quality indexes for each process and product.

- The IMA agents perform, in background and continuously, an analysis of historical and real-time data, applying data mining techniques to generate warning and guidelines to be proposed to PA and RA agents. As an example, IMA can complement PA agents, with their wider scope, to detect if the desired quality of a washing machine being produced is in

risk: a yellow warning is generated if the achievement of the desired quality is in risk, and a red warning is generated when it is not possible anymore to achieve the desired quality (even if the remaining operations will be performed with a performance of 100%). Fig. 10 illustrates a screenshot of the IMA agent, being possible to verify that two washing machines are marked with yellow colour and one with red colour.

Other self-adaptation mechanisms were implemented, tested and successfully operated in the factory plant, namely the adaptation of individual processes based on the data captured, analysed and correlated from posterior workstations.

1.4 Benefits and Assessment

The multi-agent system deployed in the real industrial production line was intensively tested during its operation, being possible to extract some quantitative and qualitative benefits.

1.4.1 Qualitative properties

The experimental tests showed that the installed agent-based system reaches several important qualitative properties:

- *Modularity and flexibility*: the use of the MAS principles simplifies the development of complex computational software applications by dividing the complex problem into simple ones. This allows achieving modularity since the system specifications are built for 4 types of agents, i.e. PTA, PA, RA and IMA, each one exhibiting a proper behaviour. For the installation in the Naples factory, several instances for each type of agent were created, each one using the same agent codification and inheriting its behaviour but customized for its particularities according to a XML file.
- *Distribution*: due to the distributed architecture provided by the MAS system, the distribution of large scale systems is easier as agents might be distributed by hardware computational resources according to the application needs, e.g., geographical dependency or processing capabilities. In the installed system, the agents were distributed by several PCs disposed along the production line.
- *Run-time and on the fly reconfiguration*: in such systems, agents can be removed, others can be added or even some modifications can be performed in the behaviour of the agents without the need to stop, re-program and reinitialize the other components (i.e. the system can continue running without any perturbation). In the installed system, this is illustrated in several ways, e.g. shutting down RA agents (e.g., the one associated to the A-Bearing station), adding new RA agents during operation (e.g., associated to functional tests boxes) or, changing the algorithms embedded in the RA agent associated to the Screwing Upper Counterweight station. These functionalities were successfully tested and validated (note that in a centralized implementation, this feature is not possible).
- *Robustness*: in case of an individual node breakdown, the system continues running without perturbation (in opposite to the traditional

centralized structures where the system collapses). In the installed system, this is illustrated by shutting down some RAs, e.g., those related to the functional tests stations, or IMA, without affecting the global behaviour of the system.

- *Adaptation*: the use of distributed control structures allows the run-time adaptation, i.e. applying local self-adaptive concepts to adapt the system behaviour according to unplanned changes. In the installed system, this feature was illustrated by the adaptation of the parameters to execute the processing/inspection operations, to customize the sequence plan of the functional tests and to customize the flow rate parameter of the on-board controller of each washing machine. Also observed in the installed system is the fast adaptation of the system to the change of washing machines models in the production sequence, and also to the introduction of new product models in the production line.
- *Scalability*: a main drawback of MAS solutions is usually associated to the agent middleware (in this case JADE) and it is related to possible delays or congestion in the communication infra-structure due to the growth of the exchanged messages (note that the increase of agents implies an increase of exchanged messages, not in a linear way, but in a more exponential manner). From the achieved experimental results, the increase from several agents to approximately 400 agents running simultaneously and having physical representation, namely connected to washing machines and workstations, some of them exhibiting weighted GUIs, didn't provoke visible degradation at this level.
- *Smooth migration*: the use of the MAS technology allows the smooth migration from old technologies/systems to new ones. This is illustrated in the installed system with the consideration of 12 workstations, from the entire production line, to be controlled by the MAS solution; a slowly and smoothly integration of the remaining workstations along the production line can be performed gradually in the time.

1.4.2 Quantitative impact

The installed agent-based system brings some significant improvements in the factory plant, namely in terms of:

- *Production efficiency*, e.g., due to the adaptation of the process parameters, the adaptation of the functional tests for each washing machine and the customization of the on-board controller parameters, based on the production history. As an example, with this approach, the functional test time for each washing machine is reduced by approximately 20%, which implies an increase of the flow in the production line or in alternative the possibility to save one testing box (and consequently save investment and energy).
- *Cost reduction*, since the earlier identification of defects or quality unconformities in the washing machines being produced can lead to a reduction of the scraps from 5% to 3%. Additionally, stopping, in advance, the production of washing machines that never reach the desired quality, contributes to reduce the waste costs and also save production time.

- *Product quality*, since most effective quality control procedures are performed along the production line, e.g., using customized testing plans in the functional tests area, and since the on-board controller of each individual washing machine is customized according to the production historical data.

1.5 Discussion

The experience gathered from the development and installation of the agent-based solution in a real industrial production line allowed to learn some important lessons.

The first one is related to the design-deployment path of multi-agent systems. The design, debugging and deployment of agent-based systems are a complex process, usually performed in an ad-hoc manner. Traditionally, the correctness of the design can only be validated after the implementation phase, leading to a very time-consuming design-implementation process that presents high rates of misunderstanding and mistakes, and, as a consequence, it is very expensive (Colombo 1998). The use of a formal methodology, based on the Petri nets formalism, allowed a rigorous specification of the proposed solution, and posterior validation and simulation during the design phase, ensuring that the model represents correctly the specifications of the real system, permitting the correction of errors and misunderstandings, and improving the control strategies before the implementation phase.

In these scenarios, the offline tests are important to ensure that errors are corrected, but they do not replace the use of online experimentation in the industrial production line, since there are situations that arise only in real industrial environments. For this reason, it is important to properly balance the negative impacts of occupying the production line for testing and the benefits of deploying the system in a real environment.

The second lesson is related to the use of agent development frameworks. In fact, the use of an agent development framework simplifies the development of MAS solutions, since it provides a set of important functionalities that support the development, debug and operation of these systems. However, these frameworks, present some drawbacks, namely the existence of a centralized node reflecting the DF (Directory Facilitator) service, the system scalability and the performance affected by the use of Java synchronized methods (Vrba 2003), which should be improved in the future.

A particular problem is the lack of compliance when deploying MAS systems to industrial environments, which is mainly a problem of standardization in the field. FIPA, which is the main standardization body in the field, made a very important job in defining the specifications to develop MAS systems, but important issues are currently missing and require further standardization work. Examples are protocols that better fits the behaviour of industrial control systems, event notification at low control level, and integration with legacy systems in a transparent manner.

The installation in a real factory plant confirmed that real industrial production environments are different playgrounds from those found in theoretical, simulated and laboratorial ones, presenting very challenging and demanding

requirements and technical constraints. As an example, industrial environments exhibit strong constraints in terms of communication infrastructure that impose additional technical problems to the implementation of MAS solutions.

The integration with legacy systems, e.g. LabView™ applications and processing stations, are very time consuming and usually developed one-of-a-kind according to the particularities of the hardware/software devices, constituting an important task that increases the complexity of the deployment process. Additionally, the equipment disposed in such industrial environments has usually close and proprietary protocols being difficult to develop the required interfaces.

At last, the successful installation of the agent-based solution in the real industrial production line producing laundry washing machines contributes to reduce the scepticism of the industry players to adopt MAS technology, in the near future, to solve the emergent problems they are facing.

1.6 Conclusions

This chapter describes the use of the MAS technology to develop and install a modular and adaptive system, integrating the quality and process control, in a real industrial production line producing laundry washing machines. The MAS system provides intelligence and adaptation to a rigid production line, allowing to improve the product quality and production efficiency.

For this purpose, it was described the behaviour of individual agents, using the Petri nets formalism, the cooperation patterns reflecting the interaction among distributed agents that leads to the implementation of the feedback control loops, as well as the technical details related to the implementation of the agent-based solution using the JADE framework.

The results achieved from the operation of the agent-based solution in the factory plant allowed to summarize a set of benefits, mainly related to the production efficiency, reflected in the customization of the plan of functional tests and optimization of process parameters, and product quality, reflected in the execution of more efficient inspection tests and customization of the on-board controller of each washing machine. The lessons learned from the gathered experience were also described.

Finally, and summarizing, the installation of the agent-based solution in the real industrial production line will contribute for a wider dissemination of the MAS technology in industry environments, reducing the scepticism of the industry players to adopt the MAS technology.

Acknowledgments

This work has been partly financed by the EU Commission, within the research contract GRACE coordinated by Univ. Politecnica delle Marche and having partners SINTEF, AEA srl, Instituto Politécnico de Bragança, Whirlpool Europe srl, Siemens AG.

References

- Bellifemine, F., Caire & Greenwood D., 2007. *Developing Multi-Agent Systems with JADE*. Wiley.
- Colombo, A.W., 1998. *Development and Implementation of Hierarchical Control Structures of Flexible Production Systems Using High-Level Petri Nets*. Manufacturing Automation Series, Meisenbach Verlag Bamberg.
- FIPA, 2013. *The Foundation for Intelligent Physical Agents*. IEEE Computer Society. Available at: <http://www.fipa.org/>. [Accessed February 8, 2014].
- Leitão, P., Rodrigues, N., Turrin, C., Pagani, A. & Petrali, P., 2012. GRACE Ontology Integrating Process and Quality Control. *Proceedings of the 38th Annual Conference of the IEEE Industrial Electronics Society (IECON'12)*, Montreal, Canada, 25-28 October, pp. 4328-4333.
- Leitão, P., Marik, V. & Vrba, P., 2013. Past, Present, and Future of Industrial Agent Applications. *IEEE Transactions Industrial Informatics*, 9(4), pp. 2360–2372.
- Leitão, P. & Restivo, F., 2006. ADACOR: A Holonic Architecture for Agile and Adaptive Manufacturing Control. *Computers in Industry*, 57, pp. 121–130.
- Leitão, P. & Rodrigues, N., 2011. Multi-Agent System for On-demand Production Integrating Production and Quality Control. *Holonic and Multi-Agent Systems for Manufacturing*, V. Marik, P. Vrba, and P. Leitão (eds.), *Lecture Notes in Computer Science*, vol. 6867, pp. 84-93.
- Murata, T., 1989. Petri nets: Properties, Analysis and Applications. *IEEE*, 77, pp. 541–580.
- Protégé, 2014. Available at: <http://protege.stanford.edu/> [Accessed February 8, 2014].
- Vrba, P., 2003. Java-Based Agent Platform Evaluation. *Holonic and Multi-Agent Systems for Manufacturing*, V. Marik, D. McFarlane and P. Valckenaers (eds.), *Lecture Notes in Computer Science*, vol 2744, pp. 47-58.
- W3C, 2004. (OWL) Web Ontology Language Reference, W3C Recommendation.
- Pechoucek, M., Marik, V., 2008. Industrial Deployment of Multi-agent Technologies: Review and Selected Case Studies, *Autonomous Agents and Multi-agent Systems*, vol. 17, n. 13, pp. 397-431.
- Gruber, T., 1995. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies*, vol. 43, n. 5/6, pp. 907-928.
- Wooldridge, M., 2002. *An Introduction to Multi-agent Systems*. John Wiley & Sons.