

PRIME as a Generic Agent Based Framework to Support Pluggability and Reconfigurability Using Different Technologies

André Dionísio Rocha^(✉), Diogo Barata, Giovanni Di Orio,
Tiago Santos, and José Barata

CTS - Uninova, Departamento de Engenharia Electrotécnica,
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa,
2829-516 Caparica, Portugal
{andre.rocha,gido,jab}@uninova.pt,
{d.barata,tp.santos}@campus.fct.unl.pt

Abstract. Presently the manufacturers are facing a challenging and important period. The markets are very different comparatively to the past, with constant changes and facing a big crisis. Hence, with these new characteristics is very important for the companies quick adaptations in order to take advantage of new business opportunities. However on the shop floor is not easy reconfigure the production lines to perform new tasks. The PRIME framework presents an agent based solution capable to rapidly reconfigure and readapt the production line using standard technology. In this paper is demonstrated the usability of this framework with three completely different technologies. With this flexibility, using PRIME it is possible perform plug and produce, reconfigurability and monitoring for all kind of technologies, not obligating the companies to reform all components in the line, avoiding external costs and stoppages.

Keywords: Multiagent systems · Standard technology · Plug and Produce · Reconfigurability · Distributed systems · Evolvable production systems

1 Introduction

The world and consequently markets are facing a global financial crisis. This crisis is one of the most challenging era in the entire history. With this new challenges, the enterprises need to re-adapt the entire way as they face these new market needs, more specifically, competitiveness, new customer needs and requirements. So, it is important for all the manufacturers, in particular Small and Medium Enterprises (SME), an effective approach, capable to allow these companies higher levels of flexibility, dynamism, and reconfigurability on the shop-floor level. As result of this new world, the manufacturers are constantly trying to solve the problem of volatility.

Nowadays the product life cycle is constantly decreasing and in some cases the companies are not able to re-adapt their production lines to new business opportunities because of the time and costs required for this re-adaptation. In order to allow

SMEs to face this problem, recently a lot of production paradigms have been proposed and developed in academic environments. However, a real response for this problem was not acquired and the problem keeps present day-by-day inside companies. The PRIME architecture [1] developed in the scope of PRIME project, funded by the European FP7 program, proposes a new approach, this new approach results in a solution to perform plug and produce on the shop-floor using any kind of computational devices. This means that to run PRIME architecture as an agent based framework [2] performing plug and produce, in a production line, it is not necessary reformulate the entire system, introducing specific controllers with a minimum computational performance to run a framework capable to offer reconfigurability, adaptability and flexibility. In this document is presented a demonstration of how it is possible run this framework, using different controllers or PLCs, in the same production line, but using the same framework.

This document is divided by six sections. In the next sections is described the PRIME contribution to cloud-based engineering systems. In section three it is possible to see a literature review about this topic. In section four is described the PRIME Multiagent Framework and how this framework can detect and plug a new resource. Section five contains three different test cases, each test case represents a detection, in this case plug, of a specific resource, using different controllers (standard PLC, DPWS controller and Windows based controller). In the last section are compiled the conclusions and further works.

2 Contribution to Cloud-Based Engineering Systems

In the last few decades the computational systems has suffered a huge improvement and development. The IT area is not an exception and as result of those improvements several cloud based solutions have been proposed for all different problems. The manufacturing systems are not an exception and in the last years several solutions have been presented to improve the performance and analyses of the systems. However, the main contributions have not resulted in valid solutions for the most of the current systems, because of the specifications and restrictions of each system, obligating to implementations very rigid and specific for each system. Without a generic way it is impossible for most of the SMEs manufacturers dispend a huge effort to implement an entire collector of data infrastructure, for its specific system. In this document is presented three different test cases using PRIME as a generic framework, capable to work with standard technology to collect all relevant data and store it into a database. Hence, this information can be used to perform data analyses, understanding the evolution of the system and possible changes. With this generic approach, the SMEs are able to quick and without unnecessary effort collect and process relevant data for the entire line.

3 Related Work

Over the last decades, the way as academics and enterprises look to the production lines has changed a lot. In this section it is described a brief study about some approaches and paradigms designed and developed under these new visions.

3.1 Agile Manufacturing

Agile manufacturing (AM) has been first proposed during the 1990s [3]. During this decade many studies and developments were introduced in this topic. A non-consensual definition for this concept was achieved but all they are basically related to responsiveness.

In [4] “Agile Manufacturing can be defined as the capability of surviving and prospering in a competitive environment on continuous and unpredictable change by reacting quickly and effectively to changing markets, driven by customer-designed products and services”.

Agile Manufacturing is a top down approach that covers the entire firm, is not a new production paradigm itself, but is more a new way to do business. However, the agility on the shop-floor level is not so easy to achieve and some new paradigms to manage the shop-floor behaviours and deal with all complexity have been proposed, such as Holonic Manufacturing Systems [5], [6], Reconfigurable Manufacturing Systems [7], [8], Evolvable Assembly Systems [9], [10] and Evolvable Production Systems [11], [12]. Those approaches bring flexibility and responsiveness to the shop-floor. To implement those kind of production paradigms, the enterprises should re-adapt the hardware in order to run intelligent entities inside each module on the line.

3.2 Evolvable Production Systems

The Evolvable Production Systems (EPS) paradigm has been first presented in the scope of EUPASS project [13]. The main idea in an EPS is a re-usage of modules in different or in the same production line constituting different systems with different valences, without any reprogramming effort. First of all it is necessary that all components are abstracted by intelligent entities. Those entities should be inserted in a society of entities (production line) and re-adapt them behaviours accordingly to the existent entities and current physical positions and status.

With this approach, the system has not a central unit to control the entire system, but distributed entities responsible to specific tasks. With this approach, it is possible introduce the concept of plug and produce. This means, plug a new resource (module) in the production line without being necessary stop the line, reconfigure the entire line and reprogramming.

In the IDEAS project an architecture was proposed, based on CoBASA architecture [14], to perform those kind of behaviours on the shop-floor level [15]. This approach is constituted by generic entities responsible to abstract different modules on the system and manage the availability of functionalities and tasks to be executed.

Although some possibilities were presented, as was described in this section, with the current economic state and needs, it is impossible for the SMEs change the entire hardware of the line to be able to run this kind of system. PRIME proposes a new generic approach, capable to run in hardware with different characteristics, fulfilling the current needs, without obligating the SMEs to change and reform the lines in order to run the previously presented approaches.

4 PRIME Multiagent Framework

The PRIME Multiagent Framework has been developed under the framework of the FP7 PRIME project. This framework has as main goal provide, in an easy way, re-configuration, adaptation, plug-ability and monitoring in a shop-floor environment.

The framework provides these behaviours using different hardware and requirements. Hence, PRIME should provide all these behaviours in all possible production lines, such as production lines composed by standard technology or production lines constituted by intelligent devices. The framework is composed by generic agents and each one has a distinct task, as represented in Fig. 1.

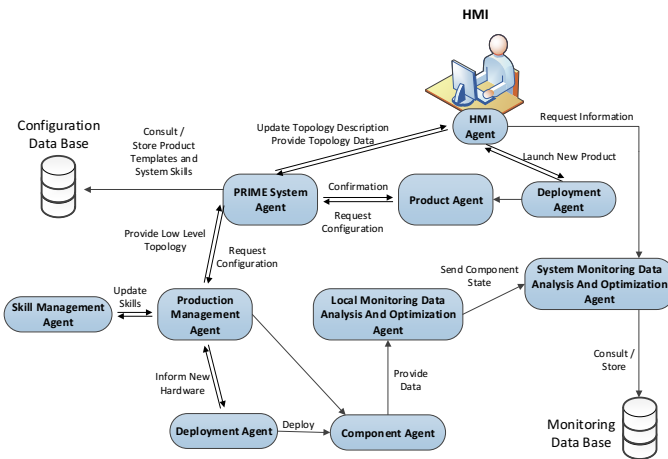


Fig. 1. PRIME Multiagent Framework

Two databases are used to store relevant information, such as performance and status data, used by the system to perform monitoring, and information related to possible configurations for the resources plugged in the system, according to the production needs, such as product variants and usage of different resources for the same tasks. More specifically, when the operator wants to produce a new variant of products, the system is able to get the most appropriated configurations that should be sent for each component and, in this way, reconfigure the entire line accordingly to the specifications of this new variant.

The proposed framework is composed by eight distinct agents with different responsibilities, some agents are responsible to abstract and monitoring the hardware

and other agents are more related to higher level actions, such as combination of different tasks that can be provided by the hardware, each task provided by each resource is described as skill that can be offered to the product. The product is described as a list of skills that must be performed by the system, in order to produce it. The PRIME Multiagent Framework is responsible to match those requested skills and reconfigure the entire line accordingly to those requirements. With this approach it is easier for the operator reconfigure the line, using the description of the necessary skills for a specific product.

- Prime System Agent (PSA) – The PSA is the higher entity in the entire framework. This agent is unique for each instantiation of PRIME, so is the central point of the entire framework. PSA is responsible to manage the PRIME Semantic Model where all the current state of the system, for example, topology and available functionalities, are stored.
- Production Management Agent (PMA) – The PMA is responsible to aggregate resources and responsibilities which are localized in the same space, allowing the aggregated area to provide higher tasks orchestrating these aggregated resources. This agent works as a tree, in order to constitute different layers of aggregations and increasing modularity and scalability of the entire systems.
- Skill Management Agent (SMA) – Each SMA works in pair with a specific PMA. The SMA is responsible to receive the existent skills in a PMA and combine, according to the defined rules, the higher level skills that the PMA can provide.
- Component Agent (CA) – The CA is the lower entity and it is responsible by abstract each physical hardware that constitutes the topology of the system. More specifically, the CA write and read data related to the execution of the entity abstracted by it. The resource has the possibility to be configured by this agent and the CA can extract relevant data as well.
- Local Monitoring and Data Analyses Optimization Agent (LMDAOA) – Similarly to the PMA and SMA, the LMDAOA works in pair with a CA. In this specific case, each CA has an associated entity of this type. This agent is responsible to receive the relevant extracted data from CA and pre-process it and analyses it, in order to define the first layer of monitoring.
- System Monitoring and Data Analyses Optimization Agent (SMDAOA) – Each instantiation of the PRIME Agent Framework runs one SMDAOA. This entity is responsible to collect all relevant data of the system. All this relevant data should be provided by each LMDAOA. With this data, it will compute all the information, store in a global database with all historical data of the system, and perform monitoring analyses. This agent provides this data to the HMIA when requested.
- Human Machine Interface Agent (HMIA) – The HMIA works as a gateway to external specific HMI. This agent as the capability to offer services which can be used by other entities to extract relevant information from the entire system. Hence, an external HMI to get relevant information, such as the current or older topologies of the system and data related to the execution of the entire sys-

tem or a specific resource, should call the service that starts the communications between the agents which provides that kind of information.

- Product Agent (PA) – When the operator wants to launch a new variant of a product or even a new product, a PA should be launched containing all information which describes this new product, such as process plan with the necessary skills. The PA triggers the configuration routines with the PSA, sending the list of necessary skills.
- Deployment Agent (DA) – In all the machines (controllers and computers), where one or more PRIME agents will be launched, is mandatory that a DA is already running in each machine and must be the first agent launched. The DA has the responsibility to deploy (launch) all the necessary PRIME agents as CAs, PMAs, Pas, among others. With this feature is possible to launch any type of PRIME agents, available, in the machines, remotely.

These classes enclose and abstracts all the generic interactions between the entities. However, the CA and DA should be extended to incorporate details about specific equipment, because these agents are directly connected to specific hardware.

4.1 Detection of New Resource (Hardware)

New physical devices, such as robots, grippers, etc. can be plugged, unplugged and re-plugged in runtime. In Fig. 2 is described all necessary interactions (messages which the agents should send and process) between the agents to provide pluggability.

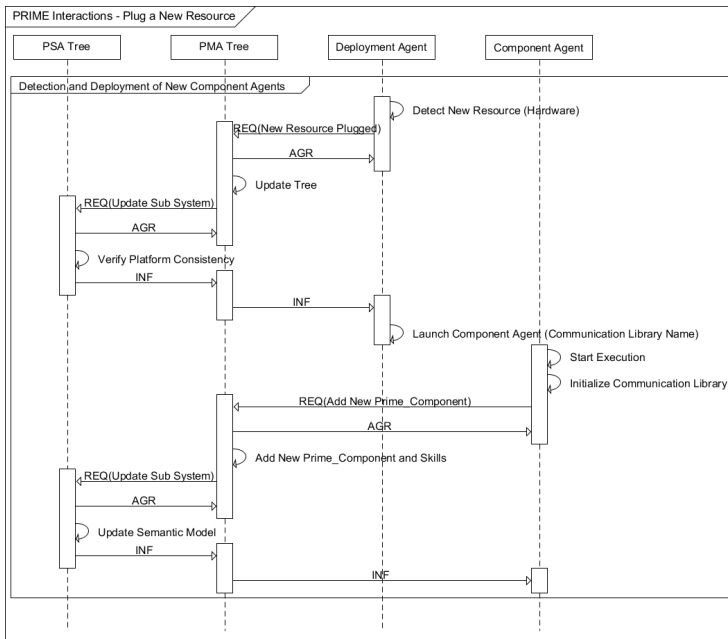


Fig. 1. PRIME Agent Interactions during a Plug

PRIME agents was developed using JADE framework [16] and all performed communications are FIPA compliant [17], more specifically is used the FIPA Request Protocol. Hence, with this functionality PRIME is able to provide plug and produce using all kind of technology.

5 Test Cases

PRIME, as was described in the previous section, is capable to detect and plug any kind of resource, with different scales of intelligence. Each device is described as one of these groups, accordingly to them capacities:

- Fully-intelligent resources: This type of resources are capable to autonomously run the necessary agents, such as CA, LMDAOA and DA. Usually this kind of resources are composed by Windows or Linux based controllers, capable to run Java environment.
- Semi-intelligent resources: These devices are able to autonomously announce the existence of new resources. The device offers the possibility to reconfigure itself but without running locally the agents.
- Passive resources: In this case, the device is a fully dummy resource, without any capacity of computation. To manage a device with these characteristics, another machine should run the necessary agents and monitoring the memory and connections with a standard PLC, responsible to control this hardware.

Three different demonstrations were created in order to test the detection of these three types of devices by the PRIME Multiagent Framework. The entities which interact with the hardware implement a generic interface that contains all specific interactions with the hardware. The CA and DA implement specific Detection and Communication libraries respectively, for each resource.

5.1 Detection and Configuration of Fully-Intelligent Resources

In this case, it has been used an intelligent device. Using a controller from Elrest, running Windows XP Pro for Embedded Systems (Fig. 3), was possible to prove that PRIME is able to manage resources controlled by such devices.

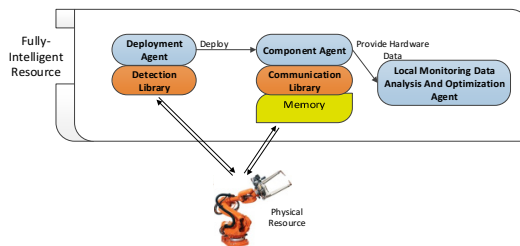


Fig. 2. PRIME Agent environment using a Fully-intelligent Resource

This kind of resources are the most suitable to integrate with PRIME, providing all the necessary information in an autonomous way. With a Windows based controller, PRIME agents can run locally, the DA, CA and LMDAOA. To detect resources controlled by these devices, when the controller executes the bootstrapping, automatically launches the DA which detects the current plugged hardware and starts the routines to launch the CA. The CA when launched, launches a LMDAOA. As the agents are running locally, when PRIME decides to configure a resource, the CA is able to write directly to the controller memory, changing the routines performed by it.

5.2 Detection and Configuration of Semi-Intelligent Resources

The second demonstration (Fig. 4) is responsible to prove the usability of Semi-intelligent resources to detect and reconfigure physical devices in a production environment by PRIME. In this was used an INICO S1000 PLC, running DPWS framework that offers services, capable to run specific routines, as reconfiguration or detection of new components.

To run this demonstration, a central machine such as a computer or a Windows / Linux based controller, should run the DPWS framework which is responsible to detect new INICO PLCs and call the services. The DA extends the libraries necessaries to launch and run the DPWS framework, and in a cyclic way checks if in this local environment was detected new resources (INICO PLCs), if yes, the DA launches a new CA in this central machine and this new CA is capable to send new configurations to the PLC, calling available services for that proposes.

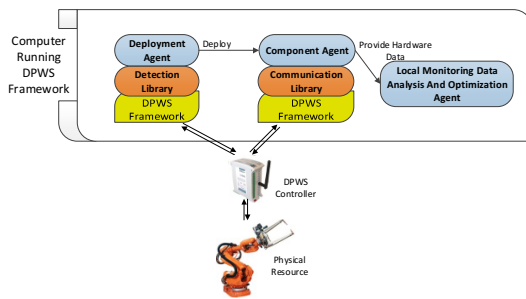


Fig. 3. PRIME Agent environment using a Semi-intelligent Resource

5.3 Detection and Configuration of Passive Resources

Although the usage of more powerful devices, as described in the previous sections, is important because of the natural evolution of manufacturing systems, standard technology still being the most common technology.

The last demonstrator, represented in Fig. 5, represents how the PRIME platform is able to manage and interact with this kind of technology.

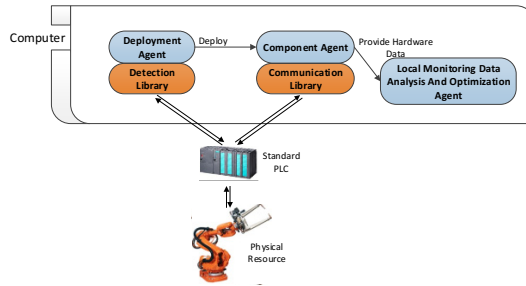


Fig. 4. PRIME Agent environment using a Passive Resource

Usually, a PLC is responsible to control a specific resource or a wrap of resources, such as robots or a conveyor belt. Hence, to integrate PRIME with the following structure of resources, it is necessary an external machine to run the agents. This external machine can be a computer or a Windows / Linux based controller, running a Java Virtual Machine and a JADE container. Inside this JADE container lives a DA responsible to detect the existence of new physical resources. In order to detect new resources, the DA extends a specific library for each case, for instance the DA can check the Ethernet ports in a local network or specific variables retrieved by the PLC. When detected a new device, a new CA will be launched to abstract the device. The CA, in this case with a standard PLC, extends a library capable to configure and consult the PLC memory, using standard protocols, such as an OPC connection.

6 Conclusions and Further Work

The PRIME Multiagent Framework was presented in previous works, with the capacity to perform reconfigurability and plug ability using standard technology. In this document was presented three different test cases that represents three different possibilities of abstractions made by the PRIME Multiagent Framework on the shop-floor using different technologies. The test cases were performed using resources controlled by Windows based controller, DPWS controller and standard PLC. In all the cases were possible to see the detection and reconfiguration of the resources using the different technologies.

With this demonstration, it is possible to say that PRIME effectively perform plug and produce, reconfigurability and monitoring using different technologies. This new proposal results in a new possibility for SMEs in a more flexible and re-configurable approach without wasting time and money adapting the entire line to the recent emerged production paradigms, which required machines with high performance.

In order to use this real advantage, it is now important to create all the tools that will be used by the operator to develop the lower level libraries, such as the detection and communication libraries. It is also important to define and create auxiliary software, capable to allow the operator an easy way to enable a module to run and to be detected by this framework.

Acknowledgments. This research was supported by the EU Frame Work 7 project PRIME (Plug and Produce Multi-agent Environment, Grant Agreement No. 314762).

References

1. Antzoulatos, N., Castro, E., Scrimieri, D., Ratchev, S.: A multi-agent architecture for plug and produce on an industrial assembly platform. *Prod. Eng.* **8**(6), 773–781 (2014)
2. Rocha, A., Di Orio, G., Barata, J., Ribeiro, L., Antzoulatos, N., Castro, E., Scrimieri, D., Ratchev, S.: An Agent Based Framework to Support Plug and Produce (2014)
3. Yusuf, Y.Y., Sarhadi, M., Gunasekaran, A.: Agile manufacturing: The drivers, concepts and attributes. *Int. J. Prod. Econ.* **62**(1–2), 33–43 (1999)
4. Gunasekaran, A.: Agile manufacturing: A framework for research and development. *Int. J. Prod. Econ.* **62**(1–2), 87–105 (1999)
5. Leitão, P., Restivo, F.: A holonic approach to dynamic manufacturing scheduling. *Robot. Comput.-Integr. Manuf.* **24**(5), 625–634 (2008)
6. Leitão, P., Restivo, F.: ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Comput. Ind.* **57**(2), 121–130 (2006)
7. Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Van Brussel, H.: Reconfigurable Manufacturing Systems. *CIRP Ann. - Manuf. Technol.* **48**(2), 527–540 (1999)
8. Koren, Y., Shpitalni, M.: Design of reconfigurable manufacturing systems. *J. Manuf. Syst.* **29**(4), 130–141 (2010)
9. Frei, R., Di Marzo Serugendo, G., Barata, J.: Designing self-organization for evolvable assembly systems. In: *Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2008*, pp. 97–106 (2008)
10. Onori, M., Barata, J., Frei, R.: Evolvable assembly systems basic principles. In: *Information Technology for Balanced Manufacturing Systems. IFIP*, vol. 220, pp. 317–328. Springer, Boston (2006)
11. Neves, P., Barata, J.: Evolvable production systems. In: *IEEE International Symposium on Assembly and Manufacturing, ISAM 2009*, pp. 189–195 (2009)
12. Ribeiro, L., Barata, J.: Re-thinking diagnosis for future automation systems: An analysis of current diagnostic practices and their applicability in emerging IT based production paradigms. *Comput. Ind.* **62**(7), 639–659 (2011)
13. Onori, M., Alsterman, H., Barata, J.: An architecture development approach for evolvable assembly systems. In: *The 6th IEEE Int. Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing, ISATP 2005*, pp. 19–24 (2005)
14. de Oliveira, J.A.B.: Coalition based approach for shop floor agility – a multiagent approach (2003)
15. Ribeiro, L., Barata, J., Onori, M., Hanisch, C., Hoos, J., Rosa, R.: Self-organization in automation - the IDEAS pre-demonstrator. In: *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pp. 2752–2757 (2011)
16. Bellifemine, F.L., Caire, G., Greenwood, D.: *Developing Multi-Agent Systems with JADE*. John Wiley & Sons (2007)
17. Bellifemine, F., Poggi, A., Rimassa, G.: JADE–A FIPA-compliant agent framework. In: *Proceedings of PAAM*, vol. 99, p. 33 (1999)