Self-Learning Production Systems (SLPS) – Optimization of Manufacturing process parameters for the Shoe Industry

Giovanni Di Orio, Gonçalo Cândido, José Barata  
CTS – UNINOVA, Dep. de Eng. Electrotécnia  
Faculdade de Ciências e Tecnologia  
Universidade Nova de Lisboa  
2829-516 Caparica, Portugal  
{gido, gmc, jab}@uninova.pt

Sebastian Scholze, Oliver Kotte, Dragan Stokic  
ATB – Institute for Applied Systems Technology  
Bremen GmbH  
28359 Bremen, Germany  
{scholze, kotte, dragan}@atb-bremen.de

Abstract—The manufacturing processes of today are caught between the growing needs for quality, high process safety, efficiency in manufacturing process, reduced time-to-market and higher productivity. In order to meet these demands, more and more manufacturing companies are betting on the application of intelligent and more integrated monitoring and control solutions to reduce maintenance problems, production line downtimes and reduction of manufacturing operational costs while guaranteeing a more efficient management of the manufacturing resources. In this scenario, the research currently done under the scope of the Self-Learning Production Systems (SLPS) tries to fill these gaps by providing a new and integrated way for developing monitoring and control solutions based on novel technologies and especially on self-adaptive, context awareness and data mining techniques. This paper introduces the research background that has driven the design of the generic SLPS architecture and focuses on the Adapter component responsible for adapting the system behaviour according to the actual operative context. The proposed Adapter architecture together with its core components are introduced as well as the generic adaptation process, or rather, the way the Adapter adapt the system behaviour to cope with the current context. Finally, to demonstrate the applicability of the SLPS methodology into real industrial context as well as the Adapter capabilities to learn and evolve along system lifecycle an application scenario is presented.

I. INTRODUCTION

The manufacturing processes of today have become extremely complex owing to the technological advances in the last three decades. The materials and processes first used to shape the products by casting and hammering have been gradually developed over the centuries, using new materials and more complex operations at the increasing rates of production and higher levels of quality, as stated in [1]. Moreover, today’s manufacturing companies are caught between the growing needs for safety, reduced time-to-market that implies short manufacturing time, minimal manufacturing costs through the efficient use of the resources and, above all, final high quality and highly customized products. In order to meet these demands, the manufacturing companies need to operate their computer controlled machines, i.e. their manufacturing environment, as efficiently and effectiveness as possible by selecting the best set of manufacturing process parameters. Determining optimal process parameter settings critically influences productivity, quality, and cost of production. Therefore, optimal manufacturing process parameter setting is recognized as one of the most important activity, as exposed in [2]. Previously, engineers used trial-and-error processes which depend on the engineers experience and intuition to determine initial process parameter settings [3]. However as stated in [4], [5], the trends in manufacturing processes complexity reinforce the idea that the processes are characterized by a multiplicity of dynamically interacting process variables making the trial-and-error process time consuming and costly affecting, in such a way, the manufacturing companies competitiveness in market sharing. Therefore as argued in [6], the trial-and-error process appears completely not suitable in the context of actual manufacturing processes. Consequently as stated in [7], the optimization of manufacturing process parameters requires new reliable methods and approaches, based on modeling activities, in order to predict the output of the manufacturing processes or in other words to predict the behaviour of the manufacturing process. The necessary step for optimizing the manufacturing process parameters is then to understand the principles that govern the manufacturing process by the developing an explicit model of the process [8] based on empirical data.

The Self-Learning approach and methodology intends to enhance traditional monitoring and control solutions by relying context awareness [9] and data mining techniques in order to select the optimum set of manufacturing process parameters for each operative context. Thus, a self-learning production system (SLPS) will be able to both perceive the context in which manufacturing process is operating and suggest the best set of manufacturing process parameters by using the relationship between the performance of the process and its controllable input parameters.

This paper presents the research and development related to the Adapter proposing its architecture for Self-Learning Production Systems (SLPS) supported by the use of Service-
Oriented Architecture (SOA) principles and technology. Pushing SOA into production control and monitoring levels is becoming increasingly appealing since it leads to the usage of a service-based communications infrastructure to build an unified and integrated solution transparently compliant along enterprise ICT levels[10], [11]. The Adapter is the component responsible for starting an Adaptation process whenever a change in the actual production system context is perceived. The use of Data Mining techniques along production system lifecycle ensures that the Adapter always calculates the adaptation that best fits the current context based on acquired history data.

The present paper explores the application of the proposed Adapter solution employed within SLPS reference architecture into a context of manufacturing process parameters optimization for shoe industry.

II. SELF-LEARNING PRODUCTION SYSTEMS

A. Motivation and Goals

The research motivation behind this work relates with the strategic objective of strengthening EU leadership in production technologies in the global marketplace by developing innovative self-learning solutions supported by a SOA infrastructure to enable tight integration of control, maintenance of production systems while improving the final product quality. This approach requires a paradigm shift to support the merging of the world of control with other manufacturing activities of the production systems so-called secondary.

The self-learning approach is intended to have a high impact on manufacturing industries and solve open questions concerning:

- Reduction of time and efforts needed for the parameterization of production lines control systems.
- High degree of flexibility in the development and installation of production control systems.
- Reduction of down times during product exchange and/or conflicts situations.
- Increasing of Overall Equipment Effectiveness (OEE), i.e. plant availability and its productivity over time.

This research initiative is driven by three disparate business cases provided by three industrial partners to enable the application of SLPS solution into real industrial environment. For each business case several application scenarios are considered and used to assess and explore the benefits of the proposed solution.

B. SLPS Generic Architecture

The generic reference SLPS architecture (see Fig. 1) [12] is constituted by two main components: the Extractor and the Adapter operating in a cooperative manner. These two together are responsible for identifying the current context under which the production system is operating (Extractor), and adapt the production system behavior, or rather the production process parameters, with the purpose of improving general system performance and indirectly improve the final product quality in face of contextual change (Adapter).

The result of the context extraction process will be a standardized meta-model gathering all context relevant information obtained by monitoring process machines and devices and necessary to frame the manufacturing operative status. This standardized meta-model will be used by the Adapter to start an evaluation process for improving production system productivity, efficiency and performance. Finally, the outcome of the adaptation activity, the Adaptation itself, is exposed to the system expert through the Expert Collaboration UI for a system expert evaluation.

Since the system response must take into account not only the particular context, but most important, the entire lifecycle behavior of both system and expert, a Learning module has been provided, containing a set of machine learning algorithms capable for extracting patterns and regularities from gathered contextual data and operator decisions over time. All processed data and knowledge generated are stored in Data Access Layer repositories for continuous evolution and evaluation. These components allow both Extractor and Adapter to access it when they need further information about current context.

The general Self-Learning production system architecture has been designed following a modular and abstract approach in order to remain hardware-independent and still compliant with a wide range of application domains by only configuring the main components.

III. SELF-LEARNING ADAPTER

The Adapter component was developed to provide system adaptations in response not only of the current system context but also considering system evolution along its lifecycle. The Adapter is described by its behaviour on top of the proposed architecture, especially showing its main modules and the interactions between them during the Adaptation Process.

The issues briefly introduced in the previous sections have been deeply studied and investigated by the authors together
with the industrial partners in order to define a set of requirements and functionalities expected to be supported by the Adapter:

- React to a change of context and provide a suitable adaptation proposal to be validated by the system expert.
- Employ the Learning Module as a mean to process large amounts of data concerning a particular context and identify the fittest adaptation proposal to be presented to the system expert.
- Detect expert decisions about system adaptation result and deploy the validated adaptation into the system.
- Manage Adaptation Repository ensuring that each adaptation process is stored for future use and analysis.
- Proactively process existing data in search in order to improve future adaptation solutions for identified contexts as well as identify system idle time to update its core knowledge about system evolution.

A. Architecture

The generic Adapter architecture is shown in Fig. 2. The core task-oriented components of the proposed architecture are the following:

- **Context Change Handler**: responsible for asynchronously handle notification events sent by the Extractor, whenever a change in context is detected. These events will be the trigger of the adaptation process.
- **Repository Extractor**: responsible for retrieving the necessary information from the Data Access Layer repositories related to the current context change. The retrieved dataset includes all the information necessary to support the adaptation process that will, in turn, determine the appropriate adaptation proposal, i.e. machines/processes parameters and/or configurations adaptation to the new context.
- **Repository Parser**: the dataset retrieved from Data Access Layer repositories contains raw information that needs to be arranged in particular way in order to be properly processed by the Learning Module. In summary, the Repository Parser creates a generic data structure that will serve as input for the Learning Module.
- **Learning Parser**: Similarly to the Repository Parser, this component will acquire the result of the Learning Module reasoning task and parse it to create a generic data object (Adaptation), which includes all the information needed by the system expert for validation issues. Furthermore, it is also responsible for receiving a complete Adaptation (including the proposal and result of the system expert validation). This information is crucial to support the accuracy of future adaptation proposals.
- **UI Comm**: handles the interaction between the Adapter and the Expert Collaboration User Interface (UI) providing a communication channel between the system expert and the SLPS deployment. This component is also responsible for informing both the UI whenever a new adaptation proposal is ready and for detecting/retrieving an adaptation that was entered into the system through the Expert Collaboration UI.
- **Adaptation Distribution**: responsible for distributing an Adaptation object instance along the Self-Learning environment after it was transferred into the real system. It will store the current Adaptation instance in the Adaptation Repository and it will inform Context Extractor that an adaptation was done in the system.
- **Proactive Learning**: This module embodies the proactive behavior of the Adapter component by performing two main tasks: the first one is an event-triggered task and the second one is a cyclic task. The major goal is to improve future adaptation proposals and exploit system idle times by running learning tasks.

Each component has a different role during the process of collaboratively providing adaptation proposals concerning the behavior and parameters of the current context. Since the Adapter is simply one brick of the overall infrastructure, it needs to interact with other surrounding modules to entirely fulfill its goals. This way the Adapter will interact with the following infrastructure modules: Extractor, Learning Module, Expert Collaboration UI and Data Access Layer.

The envisioned architecture shows a purely reactive behavior represented by the adaptation process. However, a proactive behavior has been embedded into the Adapter to save time and resources during system run-time and allow the configuration of the Adapter lifecycle evolution trend through some settable parameters. To better clarify the Adapter role within the SLPS environment and thus the interactions with the others modules as well as the rule of its own constituent components, an adaptation process will be explained in section III-B.
B. Adaptation Process

The Adaptation Process (see Fig. 3) refers to the execution of a sequence of tasks for evaluating production system parameterization consistency, which are performed every time the Extractor notifies the Adapter about a change of context. This notification represents the trigger that will drive the SLPS to adapt itself to the current context.

After being notified about a change of context, the first thing the Adapter do is to retrieve all the available information related to this new context. This task is performed by the Repository Extractor, which retrieves, from the Data Access Layer, all the according datasets and models for the current context (monitoring dataset). This collection of data is then transferred to the Repository Parser component to transform it into a generic data structure, (ReasoningInput), that structures the retrieved monitoring dataset. The ReasoningInput will be used as input for the Learning Module whenever there is a need to perform a reasoning task. A reasoning task will exploit supervised machine learning classification techniques that, based on a learning model comprising the entire lifecycle behaviour of the system, will breed new production system parametrization proposals. Subsequently, the Learning Module will deliver the reasoning task result to the Learning Parser component and create the Adaptation object instance. Specifically, at this moment, the Adaptation will include the current context dataset that has triggered the Adaptation Process, a proposal of adaptation and the final result of the adaptation. The proposal of adaptation will contain the result of the reasoning task, which is going to be shown to the system expert, while the result of the adaptation will contain the system expert final decision. This way, the Adaptation object will be transmitted through the Comm UI component to the Expert Collaboration UI that waits for the system expert to validate, modify or refuse the original adaptation proposal. The system expert decision will be stored in the Adaptation object as final result of the adaptation, and at the same time transferred to the real production system equipment. Finally, the Adaptation object will be distributed within the SLPS environment, i.e. notify the Extractor concerning the end of the Adaptation Process, save the Adaptation into the according repository and invoke the Learning Module service to update the learning model of the process with the related context information and the final result of the adaptation. Since supervised machine learning techniques are used to adapt system behaviour during its lifecycle, an externally supplied set of context instances are required to build a learning model for the particular application scenario. Once a learning model is available, an inductive machine learning process can be carry out representing the process of learning a set of rules from instances (examples in a training set), creating a classifier/estimator that can be used to generalize from new instances [13]. During the classification/regression task the choice of the specific learning algorithm is a critical step. The learning algorithm choice has been realized considering prediction accuracy and/or relative error. In this scenario several statistical comparisons were conducted between the learning algorithms coming to a final well suited solution based on the nature of each application scenario.

C. Proactive behavior

Although triggered by the Extractor whenever a change in context is detected, the Adapter is also capable of monitoring its own state during system operation in order to identify instants in time to proactively launch new learning tasks (Proactive Learning). Therefore, when a learning task is launched, i.e. whenever a learn command is sent to the Learning Module, a new learning model referring to all the stored contexts information is inferred and a Cross validation [14] is performed to analyze the capability of the model to generalize in face of an independent dataset. The result of the cross validation is then stored into an appropriate repository that can be queried by the user for retrieving statistical information useful for him to assess the quality of the adaptation proposals. Since the learning task is a time consuming activity, the proactive behaviour should optimize this process identifying inactivity periods as well as obsolete learning models. The Fig. 4 presents an overview of the Adapter proactive behavior.

Therefore, during production system operation, the Adapter
will verify the number of performed adaptation processes and the elapsed time since last adaptation for detecting when a model can be considered out-dated and/or when the system is idle.

The level of proactivity is defined through a configuration file, which can be configured by the user in order to specify the thresholds for triggering new learning tasks. The specification of the proactivity thresholds should be a good compromise between system accuracy and the processing time during the adaptation processes, since a large amount of context instances to learn implies time and processing power for building a new learning model.

IV. Application Scenarios

The SLPS solution has been designed and developed with the contribution and support of three industrial partners with distinct manufacturing background. The efforts of both academic and industrial world resulted in a methodology and platform capable to tackle some weaknesses of traditional monitoring and control solutions. The SLPS solution is assessed and validated in all application scenarios at these manufacturing companies. However, this paper presents only one application scenario.

A. Self-Learning optimization of manufacturing process parameters for shoe industry

One of the involved industrial partners addresses control systems/machines and automation systems for shoe industry. As stated in [15], [16], [17], the production and manufacturing of shoes involves a wide variety of materials and a large number of operations grouped into five major activities, namely cutting, stitching, preparation, lasting and finishing. These activities comprises a set of complex operations that are labor intensive and operator skill dependent. Taking into account these aspects, the need for automatic methodologies for the recognition of anomalous situations, that potentially may cause a line stopping or degradation of product quality has been identified by producers of machinery for shoe industry. Thus, the main objective in this business case is to enhance actual monitoring and control solutions with self-learning capabilities to allow machines to inspect statistically the condition of products and equipment, report and analyze proactively the gathered statistic values, and adapt the manufacturing process parameters in order to keep them always inside the optimum working range achieving, in this way, high adaptation of the machines to the changing conditions.

B. Early Experimental Results

To validate current SLPS platform fitness and performance, the proposed SLPS solution has been integrated into real industrial equipment and used to identify production process operative context and react to changing situations associated with variations in different parameter sets in order to improve error-prone processes (caused by humans) and reduce maintenance problems. The considered application scenario addresses an improved synchronization of the so called “mixing head” system (see Fig. 5), constituted by different valve circuits operated for metering several components.

During the production process of shoe sole, different components are mixed by synchronously acting on different non-mechanical connected valves. However, after several production cycle the valves get asynchronous due to a variety of situation such as different force requirements, different air supply and/or valve abrasion that affect the final product quality. The main objective is to automatically adjust the opening time of the valves based on the operative context of the manufacturing production system. To do this, the SLPS solution has been fed with a set manufacturing process parameters which values have been gathered by observing the production process. These set of parameters are then used by the Adapter to build a representative model of process relying on empirical data using data mining techniques. The parameters considered to build the model are the pressure and the temperature, speed frequencies of drives and pumps, proper material mix ratio and filling of materials into shoe forms.

The Fig. 6 explains the valve synchronization application scenario.
before shown a notification is sent to the Adapter and an Adaptation Process is started leading to a proposal displayed to the system expert by the Expert Collaboration UI (see Fig. 7). The final result of the Adapter reasoning activity is a prediction about the opening time of the valves that in turn is used to identify if a synchronization task is needed.

Figure 7. Valve Synchronization use case Expert Collaboration UI

First application of the presented approach in real world scenarios is pointing at promising results. The application in control systems/machines and automation systems for the shoe industry documented that the objective to enhance machines with self-learning functionalities to keep the process parameters always inside the optimum working range were fulfilled. Therefore, implementation of the proposed self-learning solution for the automatic adjustment of machine parameters based on changing context, for example changing ambient conditions, leads to minimization of errors and keeps the machine utilization high, as well as the overall product quality.

V. CONCLUSION

Current work presents a contribution to the SLPS research domain presenting a novel approach for the realization of self-learning production systems. Moreover the generic architecture of the SLPS Adapter together with a detailed description of its components and behaviour is presented. The proposed solution addresses the adaptation of critical manufacturing process parameters based on a context aware approach and data mining techniques for improving the final product quality. Further research will focus on advanced algorithms for self-learning based on extracted context to (semi-)automatically update the context model. In addition the context model itself will be addressed by further research to allow better utilization of the presented model for other companies as well as for other application domains. High complexity of data acquisition and real-time data analysis algorithms will be addressed in further research to “fully” utilize the opportunities offered by service-based self-learning systems.

ACKNOWLEDGMENT

This work is partly supported by the Self-Learning (Reliable self-learning production system based on context aware services) project of European Union 7th Framework Program, under the grant agreement no. NMP-2008-228857. This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content. This work is also supported by FCT – Fundação para a Ciência e Tecnologia under project grant Pest-OE/EEI/UI0066/2011.

REFERENCES