Self-Learning Monitoring and Control of Manufacturing Processes Based on Rule Induction and Event Processing

Daniel Metz, Sachin Karadgi, Ulf Müller, Manfred Grauer
Information Systems Institute,
University of Siegen,
Siegen, Germany
{Daniel.Metz, Sachin.Karadgi, Ulf.Mueller, Manfred.Grauer}@uni-siegen.de

Abstract **- Manufacturing enterprises are trying to cope with turbulent market situations by enhancing their existing monitoring and control of manufacturing processes. Enterprise integration within and across the enterprise can assist to realize the aforementioned goal. Further, event processing (EP) techniques can be employed to monitor and control manufacturing processes in real-time. Rules derived from stored process data using the knowledge discovery in databases process can be managed in an EP engine as event patterns. Nonetheless, rule identification is usually an offline activity whereas the control of manufacturing processes is a real-time activity. Consequently, the rule identification process should be transformed from an offline activity to an online or (near) real-time activity. In the contribution, a methodology is presented to overcome the previously mentioned drawback. Machine learning (i.e., rule induction) methods are used to automatically adapt the existing set of event patterns. The implementation of the presented methodology has been started in a casting enterprise.**

Keywords - complex event processing, rule induction, rule classification, knowledge management, real-time control, machine learning.

I. Introduction and Problem Description

Manufacturing enterprises are compelled to manufacture products with high quality and shortened lead times due to rapidly changing customer requirements, decreased life-cycle of products, and drastic variation in environmental conditions. Further, these challenges are intensified for enterprises having a high product mix with low volume production. Usually, these enterprises operate monitoring and control systems for their manufacturing processes. Nevertheless, enterprises need to enhance their existing systems to monitor and control manufacturing processes to retain their competitive advantages. Especially, enterprises strive for a higher degree of flexibility and adaptability.

The main challenge on the route to achieve the aforementioned vision is the necessity for an integrated enterprise [1]. Attempts are being made to integrate enterprise levels along horizontal and vertical direction based on ISO 15704, enterprise reference architecture [2]. Horizontal integration deals with the integration of enterprise applications (e.g., enterprise resource planning (ERP) system) or resources at a particular enterprise level. On contrary, enterprise levels are associated with different time horizons, which require vertical integration of information and knowledge. Overall, enterprise integration (EI) within and across different enterprise levels can provide a holistic view of an enterprise. Therefore, EI can be considered as a prerequisite for enhancing the monitoring and control of manufacturing processes. Further, EI can be exploited to accomplish management strategies like the real-time enterprise (RTE), support enterprise performance measurement, and enhance knowledge management (KM), among others.

During execution of manufacturing processes, enormous amount of process data (e.g., sensor readings, product quality feedbacks) is generated in (near) real-time (i.e., seconds or milliseconds) by resources located on the shop floor. In addition, operators provide necessary data related to resources, orders and products (e.g., selecting pre-defined reasons for a resource breakdown, order details during start of an order execution). This process data is utilized in different ways. First, the data is displayed in (near) real-time to enterprise members for monitoring of manufacturing processes. Second, process data is stored in relational databases for offline analysis (e.g., deriving new knowledge). Finally, the data is employed for real-time monitoring and control of manufacturing processes based on event processing (EP).

EP has become an appropriate technology for event-driven applications [3]. The knowledge (i.e., rules), derived from the offline analysis of stored / historical data using analytical techniques (e.g., data mining), can be modeled as event patterns in an EP engine. The rules can also be obtained from structured interviews with domain experts. Additionally, reactive rules can be defined, which describe (re-) actions to situations detected by analyzing the incoming process data streams. However, rule identification is an offline activity whereas controlling of manufacturing processes is a real-time activity.

In addition, today's shop floor is characterized by high automation and few employees, an employee managing multiple resources. Accordingly, the control system should be able to identify and react to situations, which are not pre-defined using the offline rules. Overall, the (near) real-time identification of rules complements the offline rule identification and enhances the performance of the manufacturing enterprise. Consequently, the existing rule identification and validation techniques need to be

 ISBN: 978-1-61208-181-6

transformed to (near) real-time activities based on the actual situations on the shop floor. As a result, the monitoring and control of manufacturing processes becomes more flexible and adaptable.

An event-driven framework has been developed at the Information Systems Institute to minimize the vertical integration gap, and monitor and control manufacturing processes based on complex event processing (CEP). This framework is now been extended to include self-learning monitoring and control mechanisms (i.e., integrate real-time control and real-time rule induction). The remaining part of the contribution is organized as follows. Section II presents research carried out in the area of manufacturing execution systems (MES), CEP, KM, and rule induction. An approach is envisaged in Section III to realize self-learning monitoring and control of manufacturing processes. The implementation of the envisaged approach has been started in an industrial scenario. This scenario is discussed in Section IV. Finally, Section V presents conclusions and outlines future work.

II. State-Of-The-Art

According to VDI 5600, an enterprise can be classified into different enterprise levels [4]. These enterprise levels are (still) inadequately integrated [5], which hinders the establishment of a holistic control of manufacturing processes [1]. Research and development has been carried out to reduce the vertical integration gap between enterprise levels. Software vendors provide MES solutions to bridge the vertical integration gap (e.g., [6][7][8]). However with these MES solutions, major issues still exist with respect to the interface between enterprise levels [5][9]. The exchange of data between enterprise control level and manufacturing execution (i.e., shop floor) is done manually or at most semi-automatically due to inflexible and proprietary interfaces [10]. Hence, standardization activities by several organizations have been performed concerning MES (e.g., [4][11][12]). Latest standardization copes with the definition of logic interfaces for machine and plant control [13].

Event driven architectures (EDA) have been introduced along with MES systems to realize the requirements of real-time monitoring and control of manufacturing processes [10][14]. FORCAM, a MES vendor, uses CEP technology as an innovative approach to monitor, analyze, and control manufacturing processes [15]. The introduction of EDA and CEP engines will assist to separate the control logic (i.e., event processing logic) from the coded application logic. Overall, this will result in an increasing flexibility and adaptability of the monitoring and control systems [16].

Rules are managed in an EP engine as event patterns. Further, the event patterns are formalized using means like event processing language (EPL) statements [10][14][16]. This knowledge is often domain specific and experts are in charge to define proper rules and statements. Knowledge management (KM) can be employed to assist experts to accomplish the aforementioned tasks [17]. For instance, the knowledge discovery in databases (KDD) process can be used to extract control-related knowledge from stored process data [17] and numerous KM tools are available to guide experts with user-friendly interfaces (e.g., [18]). However, KM tools are often utilized offline and consist of several non-trivial activities.

The ability of the control system to rapidly adapt to critical situations during manufacturing process execution is fairly limited (i.e., monitoring and control system has to be adapted manually by modifying the rule base). Classification rule induction is part of machine learning [19][20] and aims to generate a set of classification rules for a given training data set [21]. Direct methods like RIPPER [22] and CN2 [23] derive rules directly from the (process) data. In contrary, indirect methods extract rules by using classification methods like traversing of decision trees. A survey of top-down induction of decision trees classifiers has been presented [24]. Also, parallelizing of classification rule induction has been discussed [25]. The rule induction techniques have been applied in various domains like chemical process control, financial industries, diagnosis of mechanical devices, and classification of celestial objects [26].

The integration of CEP with machine learning (e.g., rule induction) has not been (extensively) explored in literature and industry [27]. A monitoring solution for application, web and database servers has been presented, which is based on the integration of CEP with the machine learning algorithm FRAHST [27][28]. Also, credit card fraud detection based on a combination of CEP with various machine learning techniques (e.g., discriminant analysis, hidden Markov models) has been investigated [29].

III. Approach For Self-Learning Monitoring and Conrtol

An overview of a self-learning monitoring and control system of manufacturing processes is depicted in Figure 1. The central idea of the system is to couple EP with machine learning techniques (i.e., rule induction). Production resources at the shop floor generate process data that denotes quality of products, parameters of resources, and performance of manufacturing processes, among others. This process data is collected by a data collection engine, which implements various industrial communication protocols (e.g., Modbus) [1]. Next, a data aggregation engine aggregates the collected process data with the data from enterprise applications (e.g., order details from ERP system) and builds tracking objects. Each tracking object represents a certain process entity (e.g., order, product) [30].

A tracking object can also be interpreted as a (complex) event by a CEP engine [31]. This CEP engine analyzes the incoming event streams (i.e., integrated process data as tracking objects) and detects (pre-defined) critical situations, and thus, monitors the manufacturing processes. The CEP engine deduces appropriate actions to control the underlying manufacturing processes in case of detection of critical situations. The action can be a combination of (i) displaying alarm messages with assistance of charts and gauges or via communication channels like emails and SMS, (ii) advising operators to modify resource parameters, (iii) manipulating process parameters in the controller of a resource.

Nevertheless, there are some instances where the CEP engine detects critical situations during process execution

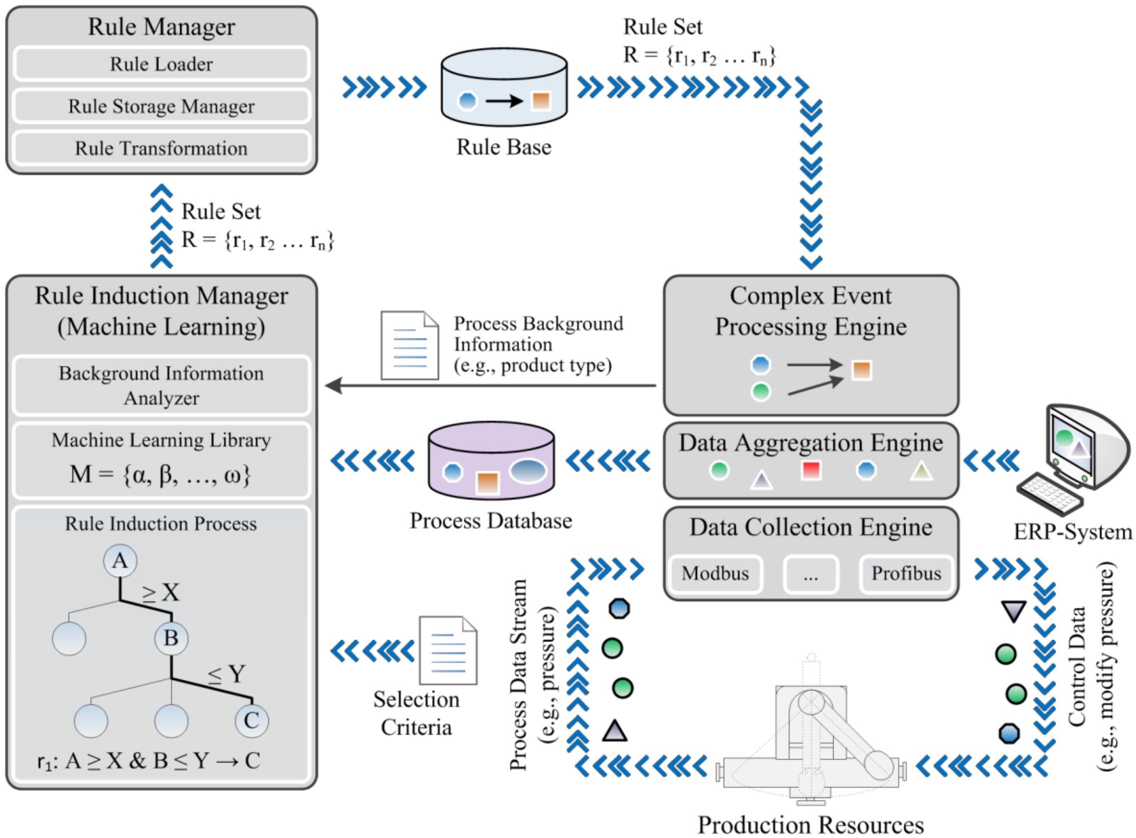

Figure 1. Overview of self-learning monitoring and control system.

(e.g., number of product rejects has exceeded a certain threshold), but cannot provide a proper suggestion to the operators to overcome the identified situations or even manipulate the resource parameters. The manufacturing ramp-up of a novel product can be an example for such an instance. In this situation, the CEP engine initiates a process in the rule induction manager (see Figure 1) and forwards current process background information (e.g., product information) describing the context of the considered manufacturing process. The aim of the rule induction manager process is to refine / improve the rule base / event patterns employed in the CEP engine. The refinement of the rule base is performed by employing machine learning techniques (e.g., decision trees).

The rule induction manager follows a sequence of activities to refine / improve the rule base. First, the process background information (i.e., process context) is analyzed and used to select a limited process data sample from the process database. For instance, the sample size can be restricted to select data for a specific product type and given time range. This step is mandatory as a huge sample size can overstrain the rule induction process (i.e., generation of rules). In addition, parallelization of classification algorithms can be considered to reduce the computation time [25].

Second, a suitable algorithm has to be selected from a rule induction library. A concise overview of rule induction techniques, which are promising candidates related to EP are listed [29]. The current research focuses on (classification) rule induction as rules are transparent and interpretable for domain experts [21]. The selection and parameterization criteria (e.g., rule accuracy, rule coverage) of a certain rule induction algorithm is defined in an XML configuration file of the rule induction manager, and thus, can be suitably modified by domain experts.

Third, rules are generated by employing the selected rule induction algorithm. This can be either performed by a direct method (e.g., RIPPER) or indirect method (e.g., traversing of a decision tree [24]). The derived rules are evaluated against the predefined criteria. Further, the previously evaluated rules can be pruned to obtain more general rules (i.e., cover more instances of the sample data set).

Finally, the induced rules are added to the rule storage (e.g., XML file format) and loaded into the CEP engine as EPL statements. The added rule complements the available rules, which might have been derived offline or online. The default action for a newly added rule is to visualize it as an alarm message because production resources should not be automatically manipulated without operator's awareness. Nevertheless, an operator can modify the generated rules and define suitable actions (e.g., directly manipulate production resources).

IV. INDUSTRIAL CASE STUDY

The aforementioned methodology for a self-learning monitoring and control system elaborated in Section III has been (partly) put into practice in a casting enterprise [32],

 ISBN: 978-1-61208-181-6

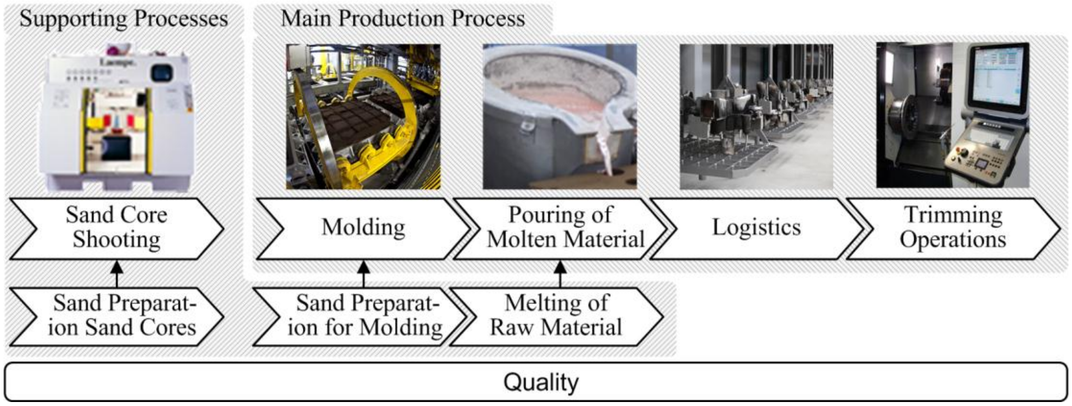

Figure 2. Simplified view of a highly automated casting process.

with the casting process as illustrated in Figure 2. The enterprise in consideration is characterized by a high mix production and low volume production (i.e., from few castings to thousands of castings per order). A highly automated molding machine is employed, which assists to realize the aforesaid characteristics.

This machine simultaneously produces upper and lower molds. The molds are manually inspected for contour and surface damages after a certain number of lower and upper molds have been produced, which is mainly due to (construction) constraints in the manufacturing system. However in case of rejection during inspection, there is a high probability that a certain number of lower molds (here: 30 molds) following the rejected molds would have similar damages.

The aforementioned situation will have a negative influence on the production performance – lower utilization of resources, material wastage and above all a declined commitment to customers. To overcome this situation, a self-learning monitoring and control of manufacturing processes is indispensable. An event-driven framework for enabling EI has been implemented using the Microsoft™ Visual Studio IDE and the .NET framework 4.0.

Real-time process data from the shop floor along with the data from enterprise applications is integrated and stored in an Oracle® 10g database. Further, the integrated data is utilized to create online tracking objects. The integrated data and online tracking objects are forwarded to process visualization clients, which display those using charts and gauges for monitoring purposes. In addition, the online tracking objects are utilized for monitoring and control of manufacturing processes using a state-of-the-art EP engine. Here, EsperTech™ EP engine is employed [33].

The EP engine uses event patterns for the detection of (critical) situations in the process data streams. By default, a rule base has been initialized with rules defined by domain experts. The main goal of the control system is to reduce the number of rejects. If this number increases without any (re-) action of the control system, a machine learning process will be automatically initiated. Classification rule induction methods have to be employed to deduce rules, which can be used to mitigate the aforementioned situation.

V. Conclusion and Future Work

Manufacturing enterprises are enhancing their monitoring and control of manufacturing processes to sustain their competitive advantages. EI can be utilized to have a holistic view of the enterprise. This EI needs to be exploited to enhance the monitoring and control of manufacturing processes. Further, research has been done to incorporate an EP engine to monitor and control manufacturing processes in (near) real-time. Here, the rules are managed as event patterns and event patterns are used to analyze the incoming process data streams. However, the rule identification and validation process is an offline activity. The existing rules do not adapt whenever there is a change in the processes' situation. Consequently, the offline activity needs to be transformed into a (near) real-time activity.

In the current contribution, an approach has been presented to identify and validate rules using rule induction techniques. On detection of certain pre-defined situations, the CEP engine triggers a sequence of steps in the rule induction manager and at the same time forwards current process background information. This sequence encompasses: (i) selecting suitable process data, which is restricted to the background information, (ii) choosing a suitable rule induction algorithm and defining selection criteria, (iii) identification and evaluation, and generalization of identified rules, and (iv) transformation of the selected rules into event patterns. The presented approach has been (partly) implemented in a casting enterprise, especially with the aim to reduce the rejection of lower molds.

Acknowledgment

Parts of the work presented here have been supported by German Federal Ministry of Economics and Technology (BMWi) as part of "Central Innovation Programme SME" (ZIM) initiative (KF2111502LL0). Also, we are thankful to our industrial partner Ohm & Häner Metallwerk GmbH & Co. KG, Germany for the opportunity to implement and validate the elaborated framework in a casting enterprise.

References

[1] M. Grauer, D. Metz, S. Karadgi, W. Schäfer, and J. W. Reichwald, "Towards an IT-Framework for Digital Enterprise

Integration," Proc. 6th Int. Conf. on Digital Enterprise Technology (DET 2009), AISC, vol. 66, Springer, Berlin, Dec. 2009, pp. 1467-1482.

[2] ISO 15704, Requirements for Enterprise Reference Architecture and Methodologies, ISO 15704:2000/Amd 1:2005, 2005.

[3] S. Chakravarthy and R. Adaikkalavan, "Provenenace and Impact of Complex Event Processing (CEP): A Retrospective View," it – Information Technology, vol. 51, no. 5, 2009, pp. 243-249.

[4] VDI 5600, Manufacturing Execution System (MES) - VDI 5600 Part 1, Verein Deutscher Ingenieure (VDI), 2007.

[5] H. Panetto and A. Molina, "Enterprise Integration and Interoperability in Manufacturing Systems: Trends and Issues," Computers in Industry, vol. 59, no. 7, 2008, pp. 641-646.

[6] J. Kletti, Ed., Manufacturing Execution System – MES, Springer, Berlin, 2007.

[7] MPDV, http://www2.mpdv.de/en/ (Accessed: 14.09.2011).

[8] S. Karnouskos, O. Baecker, L. de Souza, and P. Spiess, "Integration of SOA-Ready Networked Embedded Devices in Enterprise Systems via a Cross-Layered Web Service Infrastructure," Proc. 12th IEEE Int. Conf. on Emerging Technology and Factory Automation (ETFA 2007).

[9] B. Saenz de Ugarte, A. Artiba, and R. Pellerin, "Manufacturing Execution System – A Literature Review," Prod. Plan Control, vol. 20, no. 6, Sept. 2009, pp. 525-539.

[10] M. Grauer, S. Karadgi, D. Metz, and W.Schäfer, "An Approach for Real-Time Control of Enterprise Processes in Manufacturing using a Rule-Based System," Proc. Multikonferenz Wirtschaftsinformatik (MKWI 2010), pp. 1511-1522.

[11] VDMA 66412, Manufacturing Execution Systems (MES) - Key Performance, 2009.

[12] IEC 62264, Enterprise-Control System Integration, All Parts.

[13] VDI 5600, Manufacturing Execution System (MES) - VDI 5600 Part 3, Verein Deutscher Ingenieure (VDI), 2011.

[14] S. Karadgi, D. Metz, M. Grauer, and W. Schäfer, "An Event Driven Software Framework for Enabling Enterprise Integration and Control of Enterprise Processes," Proc. 10th Int. Conf. on Intelligent Systems Design and Applications, Cairo, 2010, pp. 24-30.

[15] FORCAM, http://www.forcam.co.uk (Accessed: 14.09.2011).

[16] O. Etzion and P. Niblett, Event Processing in Action. Manning, Greenwich, 2011.

[17] M. Grauer, D. Metz, S. S. Karadgi, and W. Schäfer, "Identification and Assimilation of Knowledge for Real-Time Control of Enterprise Processes in Manufacturing," Proc. 2nd Int'l Conf. on Information, Process and Knowledge Management (eKNOW 2010), Feb. 2010, pp. 13-16.

[18] IBM SPSS Modeler Professional, http://www.spss.com/, (Accessed: 14.09.2011).

[19] D. Michie, D. Spiegelhalter, and C. Taylor, Eds., Machine Learning, Neural and Statistical Classification. Ellis Horwood, Chichester, 1994.

[20] R. Michalski, J. Carbonell, and T. Mitchell, Eds., Machine Learning: An Artificial Intelligence Approach, Volume I . Tioga, Palo Alto, CA, 1983.

[21] P. Flach and N. Lovrac, Rule Induction, http://www.cs.uu.nl/docs/vakken/adm/RuleInduction.pdf, (Accessed: 14.09.2011).

[22] W. Cohen, "Fast Effective Rule Induction," Proc. 12th Int. Conf. on Machine Learning, 1995.

[23] P. Clark and T. Niblett, "The CN2 Induction Algorithm," Machine Learning, vol. 3, no. 4, Springer, 1989, pp. 261-283.

[24] L. Rokach and O. Maimon, "Top-Down Induction of Decision Trees Classifiers – A Survey," IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews, vol. 35, no. 4, 2005, pp. 476-487.

[25] F. Stahl, M. Bramer, and M. Adda, "PMCRI: A Parallel Modular Classification Rule Induction Framework," Lecture Notes in Computer Science, vol. 5632, 2009, pp. 148-162.

[26] P. Langley and H. A. Simon, "Applications of Machine Learning and Rule Induction," Communications of the ACM, vol. 38, 1995, pp. 55-64.

[27] P. H. dos Santos Teixeira, R. G. Clemente, R. A. Kaiser, and D. A. Vieira Jr, "HOLMES: An Event-Driven Solution to Monitor Data Centers Through Continuous Queries and Machine Learning," Proc. 4th ACM Int. Conf. on Distributed Event-Based Systems (DEBS 2010), New York, NY, USA, 2010, pp. 216-221.

[28] P. H. dos Santos Teixeira and R. L. Milidiu, "Data Stream Anomaly Detection through Principal Subspace Tracking," Proc. 2010 ACM Symposium on Applied Computing (SAC 2010), Sierre, Switzerland, 2010, pp. 1609-1616.

[29] A. Widder, R. v. Ammon, P. Schaeffer, and C. Wolff, "Identification of Suspicious, Unknown Event Patterns in an Event Cloud," Proc. ACM Int. Conf. on Distributed Event-Based Systems (DEBS 2007), Toronto, Canada, 2007, pp. 164-170.

[30] M. Grauer, S. Karadgi, and D. Metz, "Enhancement of Transparency and Adaptability by Online Tracking of Enterprise Processes," Proc. 10th Int. Conf. on Wirtschaftsinformatik, Zurich, 2011, pp. 282-291.

[31] M. Grauer, B. Seeger, D. Metz, S. Karadgi, and M. Schneider, "About Adopting Event Processing in Manufacturing," Lecture Notes in Computer Science (LNCS) 6569, M. Cezon and Y. Wolfsthal, Eds. Springer, Berlin, 2011, pp. 180-187.

[32] M. Franken, "Welcome to 21st Century," Giesserei, vol. 97, 2010, pp. 90-99 (in German).

[33] EsperTech, http://www.espertech.com/ (Accessed: 14.09.2011).

 ISBN: 978-1-61208-181-6