Development of PLC-Based Software for Increasing the Dependability of Production Automation Systems

Daniel Schütz, Andreas Wannagat, Christoph Legat, and Birgit Vogel-Heuser, *Senior Member, IEEE*

***Abstract*—This paper presents the elaboration of a concept to develop and implement real-time capable industrial automation software that increases the dependability of production automation systems by means of soft sensors. An application example with continuous behavior as it is a typical character treat of process automation is used to illustrate the initial requirements. Accordingly, the modeling concept is presented which supports application development and which is supplemented by an implementation approach for standard automation devices, e.g., programmable logic controllers. The paper further comprises an evaluation which adapts the concept for two use cases with discrete behavior (typical character treat of manufacturing automation) and validates the initially imposed requirements.**

***Index Terms*—Agent-based approaches, model-based system and software engineering, production automation.**

I. Introduction

FOR automation systems, stringent requirements regarding robustness against defects and failures exist. In manufacturing systems, interrupts resulting from defect devices are unwelcome but mostly uncritical because the treated material remains stable. In contrast, hazardous situations may arise from defect devices inside a process automation system. One of the most common sources of failures in production automation are defects of sensors [1]. There are different strategies for a control system to appropriately react on sensor faults such as shutting down the system for maintenance or forcing the controlled device to a stable state. This paper focusses on an alternative approach using redundant information from soft sensors that enable a dynamic reconfiguration of a production system during runtime.

In previous works [2], [3], the application of multi-agent systems for programming logic controller (PLC)-based automation software using IEC 61131-3 [4] has been investigated. This paper elaborates and adapts the previously developed model-based approach for developing a knowledge base for single agents in order to automatically compensate sensor faults. Furthermore, the processing of the proposed model for an implementation on decentralized PLCs enabling the dynamic reconfiguration of automation software in real time is presented. Whereas the previous works mainly focused on applications in process automation with continuous behavior, this paper evaluates the concept for applications in manufacturing automation and investigates necessary adaptations for discrete (boolean) sensors.

This paper is organized as follows. First, in Section II, an application example characterized by continuous behavior is presented in order to derive the requirements regarding a model-based approach for soft sensors. Subsequently, related work is discussed in Section III. The approach resulting from the imposed requirements is presented in Section IV. To validate the requirements and to prove its general validity, the proposed concept is evaluated using a contrasting application example characterized by discrete behavior. Section VI summarizes the approach and gives an outlook on future work.

II. Requirements for Model-Based Soft Sensors

To derive requirements regarding a model-based approach for increasing the dependability of production automation systems by means of implementing soft sensors, an application example is introduced. It contains real-time critical constraints and is appropriately complex to derive the most important requirements: a hydraulic press [5] that produces fiber boards from raw material like wood fibers and glue. The raw material is sandwiched between heat plates and steel belts that are pressurized by hydraulic cylinders. The press is composed of up to 80 press frames. Each frame consists of up to five separately controlled hydraulic cylinders that are equipped with pressure and distance sensors. The sensors are physically linked by a field bus system to several PLCs, which execute the automation software. Therefore, one PLC may control between 10 and 20 frames, i.e., 50 to 100 control loops with the same number of pressure and distance sensors. The requirements for an approach to increase the plant's dependability by soft sensors are the following:

- *(R1) Analytical dependencies between sensors*
  Analytical dependencies between different sensors (e.g., pressure and distance) have to be described inside a redundancy model using models of the plant and process (fiber material). From these analytical dependencies, soft sensor values can be calculated for redundancy. To allow the calculation of soft sensor models on PLCs with limited computing resources (see R6), the use of simplified models (for this particular plant, only partial, rigorous models exist [6]) is required.

Manuscript received December 15, 2011; revised March 30, 2012; accepted November 08, 2012. Date of publication November 21, 2012; date of current version October 14, 2013. Paper no. TII-11-1020.

D. Schütz and B. Vogel-Heuser are with the Institute of Automation and Information Systems, Technische Universität München, D-85748 Garching, Germany (e-mail: schuetz@ais.mw.tum.de).

A. Wannagat was with the Institute of Automation and Information Systems, Technische Universität München, D-85748 Garching, Germany.

C. Legat is with the Siemens AG—Corporate Technology, D-81739 Munich, Germany.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2012.2229285

- *(R2) Simplified models and their uncertainty*
  Since simplified models have to be used, the inaccuracy of the model has to be expressed by integrating possible deviations to the actual value of soft sensors, calculated online during failure-free operation. The possible deviations of real sensors have to be also integrated into the redundancy model.
- *(R3) Process and machinery constraints*
  For production and machinery parameters are constrained, e.g., desired thickness of a fiber board or maximum pressure inside the hydraulic cylinders. These constraints need to be included into the model which the implementation of the soft sensors is based on as well.
- *(R4) Sensor precisions affecting factory functions*
  Since the sensor values used in the automation software deviate from the physical state of the machinery, parameter values indicating the precision of a machinery function's execution may deviate in accordance with the precision of the sensor values used. Thus, the model has to include a calculation to determine how the uncertainty caused by imprecise sensor values affects the execution of a plant's functions.
- *(R5) Decision logic for reconfiguration of sensors*
  The concept has to contain a decision logic capable of detecting faults of real sensors—continuous (analogue) and discrete (boolean)—based on soft sensor values. This logic has to be able to evaluate all given sensor values (real sensors and soft sensors) with their particular precisions for one measurement to determine the best alternative. Finally, the decision logic needs to be able to reconfigure the automation software in order to always use the sensor values that were determined to have the highest precision.
- *(R6) Implementation on a PLC with real-time requirements*
  Based on the implementation of soft sensors from the analytical dependency model, the decision logic for broken or faulty sensors and the selection of soft sensors need to be implemented on a PLC with real-time requirements. Therefore, real-time properties of the automation system and the soft sensors need to be considered. The current state of industrial practice is the use of runtime environments conform to IEC 61131. Hence, an implementation conform to this standard is focused to increase the concept's applicability in industry. Although IEC 61499 [7] represents a promising concept for implementing flexible intelligent automation software [8], [9], an adequate hardware support in terms of interoperability with current industrial field devices is not given.

III. Related Work

The twofold nature of the contribution introduced in previous sections lead to a bisection for investigating the state of the art. The compensation of faults during operation is closely related to various concepts for flexible automation software which will be discussed in the subsequent section followed by the development of desired knowledge in order to enable dependable production control. Research conducted in engineering of flexible automation software is presented in the remainder of this section.

A. Concepts for Flexible Automation Software

Agent technology, as one specialized concept for cognitive automation software [10], is increasingly adopted for flexible automation software in order to realize different functionality required for dependable production automation like monitoring, fault diagnosis and control (see [11] and [12] for comprehensive overviews). Most applications of agent technology in industrial environments deal with different reconfiguration issues of control systems to cope with module breakdowns or structural changes of the system [13]. Long-term research activities in this domain have been carried out [14], but the compensation of sensor defects is not considered. Agent-based systems also have been applied in order to manufacture goods based on their given recipe [15] and realize distributed production planning [16], [17] by achieving higher levels of a production systems' autonomy and adaptability. Since traditional monolithic control structures do not address flexibility issues sufficiently, these approaches are often accompanied by proposals for novel control paradigms as, for instance, described in [18]. Plug and produce of self-contained modules is focused in Evolvable Production Systems [19]. Based on a description of tasks to realize a required production process, self-organizing agents controlling the system components have been proposed [20]. These approaches deal with organizational aspects of a production system on module level. Required decision options are realized through module duplication or redundant control functions offered by differing modules. Sensor defects and their impact on control and quality are not considered.

In recent years, many activities in research and industry on the IEC 61499 standard [7] for industrial distributed automation control systems have been conducted [8], resulting in an increased research on reconfigurable PLC software. An application for online reconfiguration of control behavior of inner logistic systems has been proposed in [9]. Based on an ontological situation and activity model, the automation software is reconfigured in order to react on contradictory material identification information [21]. Whereas this approach deals with redundant sensor information and its compensation, a reaction in real time cannot be provided. The reconfiguration for compensating machine breakdowns under varying throughput conditions is described in [22]. In order to reason about the execution behavior of control capabilities, nested state machines are applied to define operations necessary to adapt control behavior [23] using redundant self-contained modules.

To meet real-time requirements using agent systems, the approaches described previously divide the control level into an upper and a lower part and thus separate the underlying control task from the reconfiguration task, resulting in agents influencing the control application by varying parameter values in slow and weak real-time requirements. Some aspects in a manufacturing system, like fault-detection or error handling, are predestined to use the agents' knowledge about the system, but therefore the agents have to meet hard real-time requirements (e.g., operation cycles of PLCs). Furthermore, none of the previously introduced approaches deals with the compensation of sensor faults in real time. An agent platform for real-time operation is presented in [24] but does not consider existing standards, e.g., [4] or [7], for implementing automation software on PLCs.

In process industry, various agent-based applications, especially for critical processes, have been proposed [25]. The application of agent technologies to design fault-tolerant control systems requires detailed knowledge about used sensors and actuators [26]. Thus, strong engineering efforts (including detailed studies of installed hardware and production under control) to develop precise models hamper applications with respect to critical processes. Concerning product quality control and system damage prevention, agent-based systems have been proposed for monitoring [27] and fault detection [28]–[30] in critical processes. For monitoring and diagnosis of discrete systems, methodologies for fault identification based on supervisory control theory [31] are often applied. Ferrarini *et al.* [32] propose such an approach, which considers explicitly redundant information and is implemented on PLCs [33]. Unfortunately, the proposed algorithm is executed outside the real-time kernel of the applied PC-based automation controller. All previous approaches focus on binary decisions on the appearance of anomalies and consequently do not consider metrics for functional dependency between process values or product quality and are consequently not applicable for real-time control adjustments.

Apart from agent technology, service-oriented architectures provide another principle to encapsulate functionality in information systems. Applications have been proposed for industrial automation, such as the integration of heterogeneous devices [34], support of their deployment [35], or their composition for realizing specific processes [36]. In order to consider real-time constraints, the quality of service (QoS) concept plays an important role [37], [38]. Especially for scheduling purposes, quality models are applied to achieve desired goals despite uncertainties [39]. Whereas these approaches are characterized by rich quality models, they lack in applicability for real-time control on standard PLC platforms; likewise the service-oriented approach for timely reconfiguration of real-time systems is proposed in [40]. Furthermore, an engineering approach to describe the impact of a device's QoS aspect to the overall quality of control is not yet available.

B. Development of Models for Flexible Automation Software

An ongoing dissemination of smart mechatronic devices leads to an increasing demand for integrated modeling of hardware and control aspects. Model-based engineering techniques based e.g., on Systems Modeling Language (SysML) [41], [42], Unified Modeling Language (UML) [43], or Business process model and notation (BPMN) [44] have been proposed in this context to improve the coordination between involved disciplines through a shared language or by automatically transforming between different engineering models [45]. Since model-based engineering often comes along with techniques for automatic code generation, executable control code for PLCs can be generated [46], [47]. Hence, these approaches offer favorable features for gathering required information from application engineers but do not deal with redundant sensor information and model inaccuracy. To design and validate operation behavior of distributed automation control systems, different formal modeling approaches have been proposed [48]–[50]. The analysis of system restartability [51] and the automatic generation of adequate control sequences to bring a system back up to operation is described in [52]. All of these approaches enable the analysis of a system's control capabilities as required for robust automation control, but neither provide a breakthrough to consider sensors and actuators quality issues and their impact on control quality nor consider limited model precision.

Strong engineering efforts to develop desired models hinder applications of concepts described in the previous section. To overcome this drawback, knowledge discovery mechanisms have been proposed to discover dependencies between process variables automatically [53], [54]. A specific application of such data treatment techniques [55] is the automatic identification of soft sensors [56], [57]. Their application ranges from the estimation of unmeasured process information to sensor fault detection [58]. The complexity of the models, as, e.g., neural networks [59], used for sensor fusion concepts is recognized as well as the need for the integration of real-time aspects of the resulting sensor information. In consequence, Luo *et al.* in [60] state that the limiting factor are the computational resources. However, an integrated approach of soft sensor design for automation software does not exist. Summarizing this section, approaches of different research domains show varying strengths and weaknesses but an approach which balances between all required aspects has not been available until now.

IV. Proposed Modeling Approach

In Sections V and VI, the requirements R1 to R6 will be discussed and solutions for process automation scenarios (continuous behavior) will be proposed.

A. Analytical Dependencies Between Sensor Values (R1)

To fulfil the requirement for a description of analytical dependencies between sensor values that use simplified models of the machinery and the process, e.g., the hydraulic press and the processed fiber material from the application example, a directed graph is proposed.

In this redundancy model, each labelled node represents a measuring point and is equipped with either a real sensor or a predefined soft sensor as a value source. A quality value at each labelled node represents the accuracy (standard deviation) of the measured or calculated value. The edges of the graph describe analytical dependencies between measurement points used to calculate soft sensor values at runtime. The direction of an edge indicates where appropriate values to calculate a soft sensor can be collected. If more than one sensor is required to calculate a soft sensor, placeholder nodes (black dots) must be used. It is required that no outgoing edges are attached to these nodes. It is further imposed that only one of the soft sensors (analytical dependencies) connected to such node may be used at the same time. Thus, cycles within the graph are avoided and simultaneous use of interdependent soft sensors can be excluded.

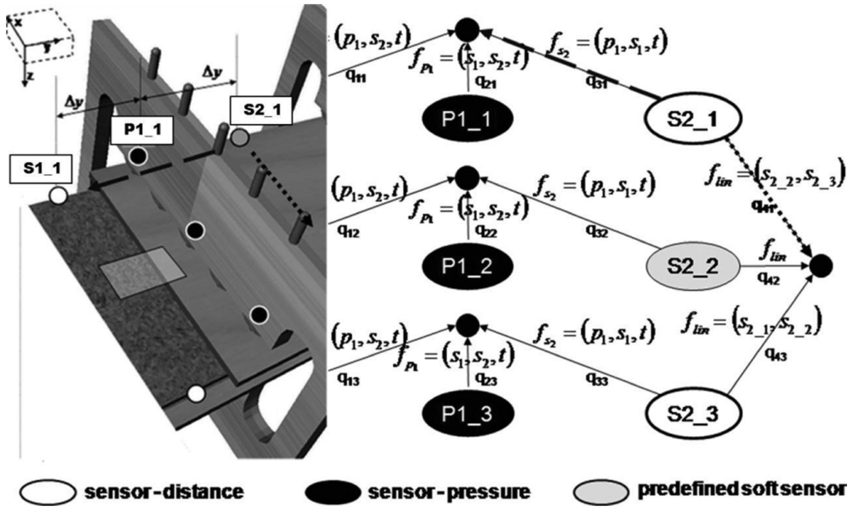

Fig. 1. Redundancy model of the application example hydraulic press [2].

The presented approach provides that an engineer with knowledge regarding a the plant and its production process defines the analytical dependencies and describes the redundancy model. Mechanisms for knowledge discovery like presented in [53] and [54] could be used to support this design process. The directed graph is not necessarily coupled to a plant's topology or characteristics of the production process. However, to gain a better performance of the implementation (see Section IV-F), it is demanded that a plant's sensors are divided according to the functions of the entities (agents) of the later automation software and one directed graph is modeled for each agent.

Application Example: In the application example, the directed graph (Fig. 1) describes the analytical dependencies between the sensors (distance and pressure) of one frame and uses a simplified model of the fiber material. It is based on the simplifying assumption that the dependency between the thickness and the applied pressure can be described by a spring constant as the elasticity of the fiber material. The time-delay, caused by the motion of the material and the distance between the sensors, is considered by using values recorded during runtime.

B. Simplified Models and Their Uncertainty (R2)

Based on the analytical dependencies described by the redundancy model, soft sensors can be calculated under the prerequisite that two aspects are considered. First, the simplified assumptions made for the analytical dependencies within the redundancy model (R1) lead to an inaccuracy and by that to an uncertainty of the calculated soft sensor values. Second, the lack in precision of real sensors, i.e., the possible deviation of the measurement. The latter can be extracted from a sensor's data sheet or determined by reference measurements if applicable regarding structural (mechanical) limitations.

These aspects are considered by a quality value (Q) for each node (measurement point) and a quality factor (q) for each edge of the redundancy model. Both of the quality values and the quality factors respectively represent the standard deviations of the measurement or the analytical dependency. Furthermore, the redundancy model considers the calculation of the quality of a soft sensor value based on the quality value of its input variables (measurement points) and the quality factor of the analytic dependency (edge) used for its calculation. For example, the quality value Q of a soft sensor V1 can be calculated by a function $Q_{V1} = f(q_{V1}, Q_{R1}, \ldots . Q_{Rn})$. Hence, a change in the quality value of an input sensor immediately leads to a loss of precision of all depending soft sensors.

C. Process and Machinery Constraints (R3)

The proposed engineering model distinguishes between two classes of constraints for plant parameters: constraints derived from limitations of the machinery and constraints resulting from production requirements. Limitations of the machinery describe boundaries, which must not be violated for not endangering the mechanical system's components ($\mathrm{Mech}+$ and $\mathrm{Mech}-$ in Fig. 2) as e.g., the mechanical parts and the sensors and actuators used. The production constraints describe technical specifications that ensure the later product quality.

Application Example: The fiber board pressing process is constrained by the material thickness in each pressing frame ($\mathrm{Proc}+, \mathrm{Proc}-$ in Fig. 2) and the pressure profile that is needed for the appropriate production of a board.

The production constraints usually allow smaller latitude than the constraints of the machinery. However, a violation of these constraints endangers the quality of the product and does not lead to any hazardous situations for men or machinery.

D. Sensor Precisions Affecting Factory Functions (R4)

Due to the uncertainty of both—available sensors (real sensors and soft sensors) and the used analytical dependencies—neither the exact values of the parameters constrained by the production or the machinery can be determined nor a set point for these parameters can be reached precisely. With the working points of plant parameters defining a correct operation, this loss in precision has to be considered by the automation software. Thus, with the precision of available sensors described in a redundancy model and constraints for plant parameters given by the machinery boundaries and production requirements, the dependency between both has to be described

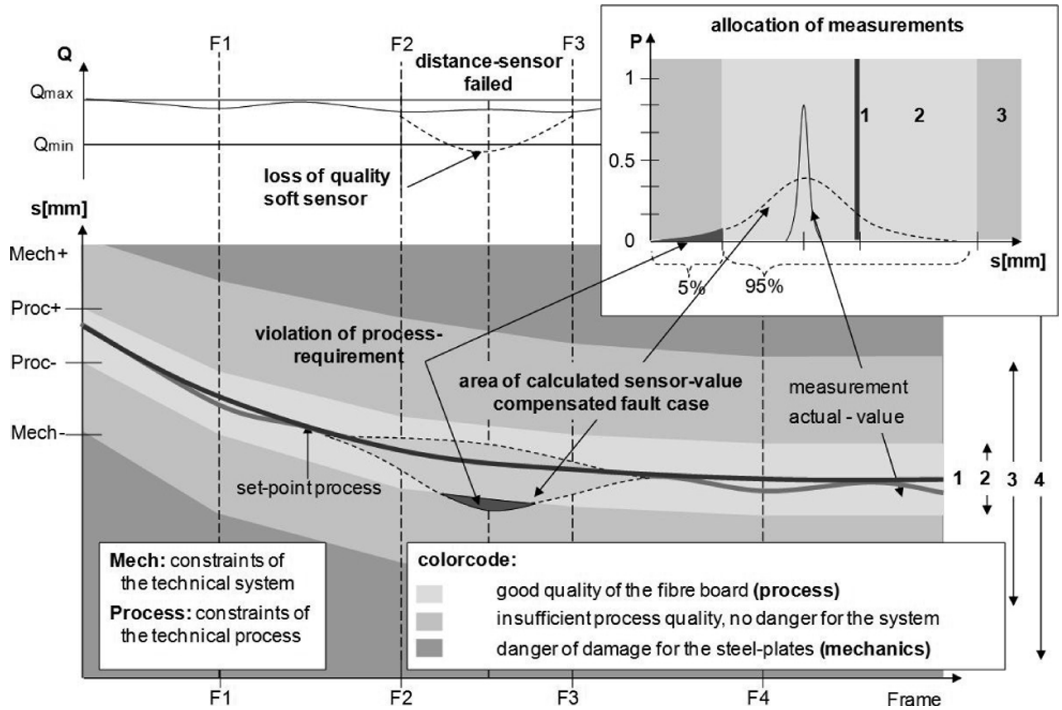

Fig. 2. Constraints of the hydraulic press (continuous process) [3].

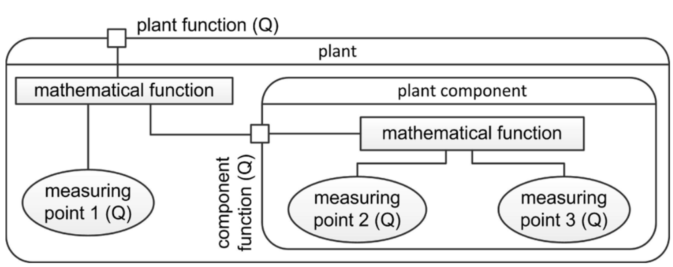

Fig. 3. Example of a tolerance model.

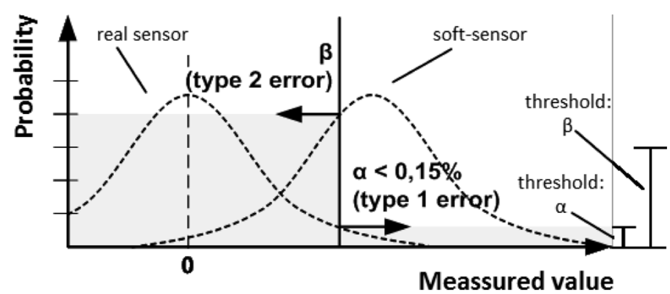

Fig. 4. Probability of error detection for different error types.

in a tolerance model, i.e., how the precision of the sensor values affects the execution of the factory functions.

During runtime, the tolerance model is calculated based on the most precise sensors currently available to verify that neither production constraints nor machinery limitations are violated. This is a prerequisite to continue a machine's operation in case of soft sensors being used.

The tolerance model can be compared with a fault tree, which describes the dependency between the precision of machinery (component) functions and the sources used for sensor values (soft sensors and real sensors) in a hierarchical, modular structure (Fig. 3). Because continuous values are used for the precision of the sensors, additionally to the concepts of a discrete fault tree, the effect of uncertainty of sensor information on the quality of a factory or machinery function is calculated, e.g., the thickness of the fiber board.

i) Application Example: In case of the hydraulic press, the precision of the plant's function, i.e., the thickness of the board, has to be evaluated. Therefore, all process parameters and sensor values regarding the modification of material properties (e.g., applied pressure or applied heat) are relevant and have to be used in the tolerance model. Hence, by calculating the distance sensors taking into account their standard deviation it is possible to calculate the probability of a violation of machinery boundaries or process requirements.

E. Decision Logic for Reconfiguration of Sensors (R5)

Fault detection based on the redundancy model uses different sources of information to identify a faulty sensor. In case of a measuring point, these are either multiple available hardware sensors (hardware redundancy) or measurements which can be used to calculate a value at a given measurement point using analytical dependencies (soft sensors). If at least three information channels are available, a defective channel can be identified by a majority vote. Due to the lack of precision, these values cannot be assumed identical even in case of absence of errors. Consequently, a threshold to recognize an error has to be defined corresponding to the expected precision of the values by the application engineer. If the threshold is too small the risk of detecting an error without a fault rises (type 1 error). If the threshold is too large the risk of not detecting an existing error rises (type 2 error).

	S1,1	S1,2	S1,3	P1,1	P1,2	P1,3	S2,1	S2,2	S2,3
S1,1	x_s1,1	x_virt_ s1,1 (1)		x_virt_ s1,1 (2)			x_virt_ s1,1 (3)		
S1,2	x_virt_ s1,2 (1)	x_s1,2			x_virt_ s1,2 (2)			x_virt_ s1,2 (3)	
S1,3		x_virt_ s1,3 (1)	x_s1,3			x_virt_ s1,3 (2)			x_virt_ s1,3 (3)
P1,1				x_p1,1			x_virt_ p1,1 (1)		
P1,2					x_p1,2			x_virt_ p1,2 (1)	
P1,3						x_p1,3			x_virt_ p1,3 (1)
S2,1				x_virt_ s2,1 (1)			x_s2,1	x_virt_ s2,1 (2)	
S2,2					x_virt_ s2,2 (1)		x_virt_ s2,2 (2)	x_s2,2	
S2,3						x_virt_ s2,3 (1)		x_virt_ s2,3 (2)	x_s2,3

Fig. 5. Redundancy matrix for one frame of the hydraulic press with P pressure sensors and S distance sensors [2].

Because of the standard deviations of the involved measurements (real sensors and soft sensors) being known from the redundancy model, hypothesis testing can be easily used to calculate the probability of an error as well as of incorrect decisions with a minimum of computing resources being required. By defining the threshold of an type 1 error (α) or type 2 error (β), it is possible to determine the sensitivity of the error detection (compare Fig. 4). Based on the specification of an type 1 error, the automation software can calculate the probability of an type 2 error. If this is also below the threshold, an error is detected. If a faulty sensor is detected the soft sensor described by the redundancy model, that currently has the lowest deviation can compensate this fault.

As stated by requirement R4, the precision of sensors used by the automation software directly affects the precision of the implemented machinery functions that use these sensors' values. Using the calculation described by the tolerance model the normal distribution for the precision value of a function regarding process parameters, e.g., thickness of the fiber board in the application example, can be calculated. Given the sharp boundary values from process requirements and plant constraints (requirement R3), the decision logic proposed above can be used to determine the probabilities of type-1 and type-2 errors for a violation of these boundary conditions. Based on the results of such calculations, the automation software may choose a correct distance between the set value for a process parameter and its boundary values.

F. *Implementation on a PLC With Real-Time Requirements (R6)*

A concept that enables the detection of sensor faults and their compensation by the use of soft sensors needs to be implemented on the same real-time capable devices that are used to control the field level of a production plant (usually PLCs). Therefore, a concept for implementing these models and decision logic within software agents using the IEC 61131-3 languages has been developed. The architecture is derived from the mechanical structure of a plant and is implemented by a hierarchy of IEC 61131-3 *Function Blocks (FB)*, i.e., each plant component is controlled by one software agent realized as a FB. For each soft sensor modelled within the directed graph, an FB is implemented in an agent's code. The FBs of soft sensors implement the calculation of the measurements and quality (standard deviation). This quality value also contains the deviations that result from real-time properties of the automation system, e.g., time aspects of the real sensors used for the calculation (see Section V for examples). Since simplified models for the soft sensors are used, such soft sensor implementations usually contain less than ten lines of code. As long as a real sensor is operating, the deviation between measured and calculated value is recorded to quantify the quality of the soft sensor. FBs of real sensors manage the sensor itself and may be used to preprocess measured input values, e.g., compressed sensor data [61].

The directed graph is implemented as an array (square matrix) of pointers (Fig. 5). The columns and rows are labeled with references to all nodes. The pointers on the main diagonal reference the measurement and quality (standard deviation) of real sensors. Soft sensors are placed in the array according to the real sensor they substitute (column) and one of the real sensors used as input (row). Each PLC cycle, the array is updated with the values and qualities of the new measurements. Therefore, the FB of each soft sensor and real sensor has to be executed in every PLC cycle. The structure of the array allows the reuse of one algorithm to compare the available sensors for a measurement, to detect faults and to choose the best value currently available. Due to the segmentation of a plant's sensors into multiple directed graphs, the resulting redundancy matrices are condensed and empty cells inside a matrix that would produce overhead and cause poor performance of the implementation are minimized.

With this matrix implementation of the available sensors and their particular quality values, the proposed decision logic for a fault detection algorithm can be realized very easily within

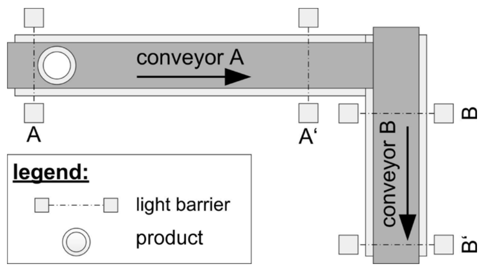

Fig. 6. Demonstration scenario: laboratory plant (discrete behavior).

an agent's implementation and executed in each PLC cycle. For each measuring point (column of the array), the agent chooses the measurement (soft sensor or real sensor) with the highest quality value. If a fault of the real sensor is detected by the decision logic (Section IV-E), its quality value will be forced to zero and the soft sensor with the now highest quality value will be chosen to replace the real sensor. Since each cell of an agent's array is evaluated regarding possible redundancies, the computing resources required for the decision logic are proportional to the array size and therefore proportional to the square of the number of real sensors. In addition, the required computing resources increase linearly with the overall number of FBs implemented for soft sensors and real sensors.

V. Evaluating the Benefit in Dependability

The requirements regarding the concept to increase a production system's dependability have been derived from a hydraulic press with continuous behavior used as an application example for process automation. As a first step to investigate the general validity of the proposed concept, it has been applied to a lab model (Fig. 6) that—in contrast to the hydraulic press—is characterized by a discrete behavior. Two connected conveyors are considered, each containing two light barriers for the detection of a product's position and a velocity sensor.

To evaluate the applicability of the proposed concept, it has to be investigated if the standard deviation of soft sensors can be used, as it is required by the concept. Two scenarios have been selected. In scenario 1, the transport of products between the light barriers A and A' is considered and a soft sensor estimating the measurement of light barrier A' is implemented. Accordingly, in scenario 2 the transport of products between the light barriers A' and B is considered and a soft sensor estimating the measurement of light barrier B is implemented. According to requirement *R1* (redundancy model), a simplified model ($s = v{\cdot}t$) was used to describe the analytical dependency between the particular light barriers and the velocity sensors of the conveyors. For each scenario, 30 test runs were executed with four different velocities (0.076 m/s; 0.11 m/s; 0.15 m/s; 0.19 m/s) during a nonfailure operation of the light barriers. The automation code implementing the soft sensors was deployed on an embedded PC providing an IEC 61131-3 runtime.

The processing time of a real sensor value t_{ProcReal} can be approximated as depicted in

$$t_{\mathrm{ProcReal}} = t_{\mathrm{del}} + t_{\mathrm{trans}} + t_{\mathrm{cyc}}. \tag{1}$$

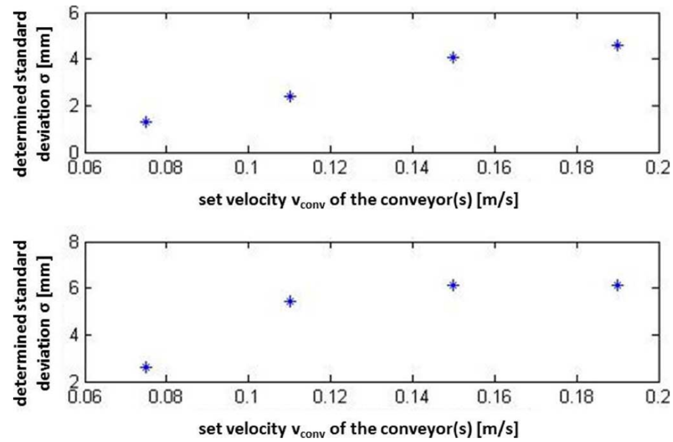

Fig. 7. Standard deviations of soft sensor in scenario 1 (top) and 2 (bottom).

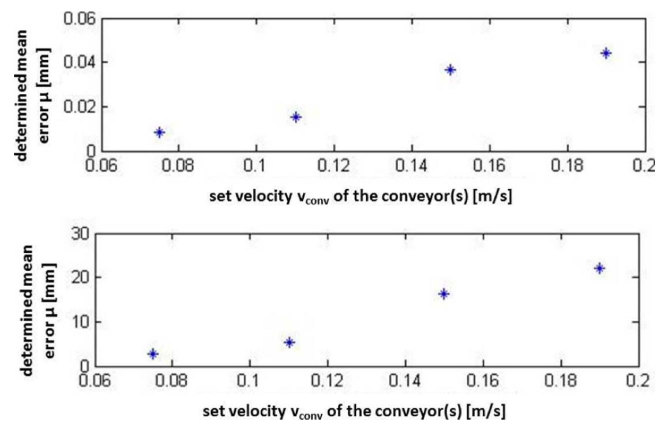

Fig. 8. Mean errors of soft sensor in scenario 1 (top) and 2 (bottom).

It relates to the delays t_{del} caused by the input terminal of the automation hardware (i.e., filtering time of digital inputs and conversion time of analogue inputs respectively), the transmission delay t_{trans} caused by the used fieldbus, and the cycle time t_{cyc} of the PLC task. In the application example, the filtering time of the input terminals was 10 μs; the transmission delay and the cycle time added up to 10 ms. During the test runs, the values of the working real sensor and the implemented soft sensor were compared and the error of the soft sensor recorded by the automation code itself. From the records, the standard deviation (Fig. 7) and the mean error (Fig. 8) of the soft sensors were derived. The results show that the simplified models (*R1*) on which the soft sensors were based on, are appropriate for the implementation but also that the standard deviation of a soft sensor must be considered. This value indicating the quality of the provided software redundancy can be adapted by the FB of a soft sensor during nonfailure operation of a plant or determined by experiments similar to the presented demonstration scenarios.

Therefore, requirement *R2* (simplified models and their uncertainty) has been proven to be valid for production systems with discrete behavior (manufacturing automation). Further, in this simple application example, the precision of the plant's function (precise transport of products) can be directly determined from the precision of the available measurements (real sensors and soft sensors) for the light barriers. In this use case, the correct detection of the product positions exists. Thus, requirements *R3* (plant and process constraints) and *R4* (sensors precisions affecting factory functions), have been proven to be valid for manufacturing applications, too.

The proposed decision logic can be used for the detection of sensor failures, but two adaptations have to be made. First, the decision logic cannot be executed in each PLC cycle but only when a product passes the light barriers, i.e., measurements of the real sensors exist. This indicates opportunities for event-triggered implementations in IEC 61499 [62]. Second, to apply the proposed concept for hypothesis testing in the hydraulic press, the measurements of the continuous sensors have been assumed normally distributed regarding their value range. Since the measurements of Boolean sensors in the conveyor example are temporarily distributed, a normal distribution regarding time is assumed. However, with the velocity of the conveyors given, the mean error and standard deviation of product positioning can be calculated (Figs. 7 and 8).

Furthermore, in the example of the hydraulic press, up to three different soft sensors were used for fault detection and compensation of a physical sensor. For the conveyor system, only one soft sensor is available for this purpose. For a qualified detection of sensor faults in this scenario, further information, e.g., based on the material flow, has to be considered for validating the correct behavior of a soft sensor or the real sensor used for its calculation. For example, in the presented demonstration scenario (Fig. 6), the soft sensor for light barrier A' can be validated when a product passes light barrier A, which is used for its calculation. In conclusion, although requirement *R5* (Decision logic for reconfiguration of sensors) can be considered to be valid for both conveyor use cases, the proposed decision logic needs to be adapted.

Nevertheless, in case of a failure of a real sensor, reconfiguration measures can be applied by the automation software during runtime, i.e., using the implemented soft sensor as an alternative source of information for the measuring point. To uphold the precision of measurement in case of a real sensor failure, further reconfigurations of the automation software can be carried out. In the test scenarios, for instance, these reconfigurations may consist of reducing the conveyor velocities according to the maximum velocity at which the precision of the product positioning fulfils the process requirements.

Additionally, the experiments showed for both use cases that the standard deviations as well as the mean error of the implemented soft sensors correlate with the set velocity of the conveyors. This is due to the fact that the precision of a soft sensor is affected by the processing time of the real sensor value which is used for its calculation. The positioning error $\mu_{\rm soft}$, which has been determined between light barrier and corresponding soft sensor, can be approximated as described by

$$\mu_{\rm soft} = v_{\rm conv} \cdot t_{\rm ProcReal} + f(v_{\rm conv}). \tag{2}$$

considering the conveyor velocity $v_{\rm conv}$, the processing time $t_{\rm ProcReal}$ of the velocity sensor [cf. (1)] and the function $f(v_{\rm conv})$, which represents the error of the soft sensor's simplified model, e.g., caused by slippage of the conveyor belt.

Hence, the measured mean error and standard deviations implicitly consider the real-time properties of the automation system. Therefore, the proposed concept fulfils requirement *R6* (Implementation in a PLC with real-time requirements) for the application example.

VI. Conclusion

In this paper, a modeling approach was evaluated describing the information necessary for a real-time implementation of single software agents that automatically compensate sensor failures in production automation. Therefore, previous works [2], [3] supporting the application of multi-agent systems inside PLC-based automation software have been investigated and adapted. The initial requirements were presented which have been derived using a real industrial hydraulic press as an application example demonstrating a continuous process. According to these requirements, a modelling approach and notation were presented that support the development of soft sensors, an implementation of software agents, and a decision logic to detect and compensate sensor failures during runtime on a PLC with IEC 61131-3 language elements.

This approach which originated from the previous works was evaluated by experiments using a use-case with two scenarios from manufacturing automation to investigate the applicability to systems with discrete (Boolean) sensors and identify necessary adaptations. The experimental results showed that the approach, which was developed according to requirements derived for a process automation system with mainly continuous sensors, can also be applied to use cases with discrete characteristics and boolean sensors like in manufacturing automation. It was recognized that, for scenarios in which only few soft sensors are available, additional information on a production system needs to be integrated into the proposed models. Furthermore, since properties of a plant's products may have to be used, the analytical dependency described for one soft sensor might not hold for the whole product range. Hence, for plants with strongly varying products, concepts for dynamic soft sensors that consider the product currently produced need to be integrated. However, the results showed that, for the particular scenarios, in case of a sensor fault, the precision of the used soft sensor can be increased by a reconfiguration of the production. Potential for an integration of the proposed concept is given because by reconfiguring the production process, software agents are able to fulfil modelled requirements from the plant or the process even if soft sensors with a lowered precision are used.

One the one hand, current projects are optimizing tool support for application engineers familiar with PLC programming and IEC 61131-3. Regarding the soft sensor models, the use of ontologies as a basis for knowledge representation is investigated to provide additional information for fault detection and compensation as well as to allow further environmental adaptation during runtime. On this occasion, the use of Bayesian networks will be investigated for representations of the proposed directed graph. Further, the application of the approach and the tool will be evaluated by application engineers from machine and plant manufacturing companies.

REFERENCES

[1] M. Cong, J. Zhang, and W. Qian, "Fault diagnosis system for automated assembly line," in *Proc. IEEE ICIPS*, 1997, pp. 1478–1482.

[2] A. Wannagat and B. Vogel-Heuser, "Increasing flexibility and availability of manufacturing systems—Dynamic reconfiguration of automation software at runtime on sensor faults," in *Proc. 9th IFAC IMS*, 2008, pp. 381–386.

[3] A. Wannagat and B. Vogel-Heuser, "Agent oriented software-development for networked embedded systems with real time and dependability requirements the domain of automation," in *Proc. 17th IFAC World Congr.*, 2008, pp. 4144–4149.

[4] *IEC 61131*, Second ed. : International Electrotechnical Commission Std..

[5] B. Vogel-Heuser*, Handbook of Automation*. New York: Springer-Verlag, 2009, ch. Automation in wood and paper industry.

[6] H. Thoemen, C. R. Haselein, and P. E. Humphrey, "Modeling the physical processes relevant during hot pressing of wood-based composites," in *Holz als Roh- und Werkstoff*, 2006, vol. 64, pp. 125–133.

[7] *IEC 61499 Function Blocks—Part 1: Architecture*. : International Electrotechnical Commission Std..

[8] V. Vyatkin, "IEC 61499 as enabler of distributed and intelligent automation: State-of-the-art review," *IEEE Trans. Ind. Inf.*, vol. 7, no. 4, pp. 768–781, Nov. 2011.

[9] W. Lepuschitz, A. Zoitl, M. Vallée, and M. Merdan, "Toward self-reconfiguration of manufacturing systems using automation agents," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 41, no. 1, pp. 52–69, Jan. 2011.

[10] D. Bruckner, H. Zeilinger, and D. Dietrich, "Cognitive automation—Survey of novel artificial general intelligence methods for the automation of human technical environments," *IEEE Trans. Ind. Inf.*, vol. 8, no. 2, pp. 206–215, May 2012.

[11] R. W. Brennan, "Toward real-time distributed intelligent control: A survey of research themes and applications," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 37, no. 5, pp. 744–765, May 2007.

[12] P. Leitão, V. Marik, and P. Vrba, "Past, present, and future of industrial agent applications," *IEEE Trans. Ind. Informat.*, vol. PP, no. 99, pp. 1–1, 2012.

[13] P. Vrba and V. Marik, "Capabilities of dynamic reconfiguration of multiagent-based industrial control systems," *IEEE Trans. Syst. Man Cybern. A, Syst. Humans*, vol. 40, no. 2, pp. 213–223, Feb. 2010.

[14] P. Vrba, P. Tichy, V. Marik, K. H. Hall, R. J. Staron, F. P. Maturana, and P. Kadera, "Rockwell automation's holonic and multiagent control systems compendium," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 41, no. 1, pp. 14–30, Jan. 2011.

[15] Y. Alsafi and V. Vyatkin, "Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 26, pp. 381–391, 2010.

[16] M. Pechoucek, M. Rehak, P. Charvat, T. Vlcek, and M. Kolar, "Agent-based approach to mass-oriented production planning: Case study," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 37, no. 3, pp. 386–395, Mar. 2007.

[17] A. Lüder, A. Klostermeyer, J. Peschke, A. Bratoukhine, and T. Sauter, "Distributed automation: Pabadis versus HMS," *IEEE Trans. Ind. Inf.*, vol. 1, no. 1, pp. 31–38, Feb. 2005.

[18] A. W. Colombo, R. Schoop, and R. Neubert, "An agent-based intelligent control platform for industrial holonic manufacturing systems," *IEEE Trans. Ind. Electron.*, vol. 53, no. 1, pp. 322–337, Jan. 2006.

[19] M. Onori and J. Barata, "Evolvable production systems: New domains within mechatronic production equipment," in *Proc. IEEE ISIE*, 2010, pp. 2653–2657.

[20] R. Frei and G. D. M. Serugendo, "Self-organizing assembly systems," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 41, no. 6, pp. 885–897, Jun. 2011.

[21] M. Vallée, M. Merdan, W. Lepuschitz, and G. Koppensteiner, "Decentralized reconfiguration of a flexible transportation system," *IEEE Trans. Ind. Inf.*, vol. 7, no. 3, pp. 505–516, Aug. 2011.

[22] M. Khalgui, O. Mosbahi, Z. Li, and H.-M. Hanisch, "Reconfiguration of distributed embedded-control systems," *IEEE Trans. Mechatron.*, vol. 16, no. 4, pp. 684–694, Aug. 2011.

[23] M. Khalgui and H.-M. Hanisch, "Reconfiguration protocol for multi-agent control software architectures," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 41, no. 1, pp. 70–80, Jan. 2011.

[24] S. Theiss, V. Vasyutynskyy, and K. Kabitzsch, "Software agents in industry: A customized framework in theory and praxis," *IEEE Trans. Ind. Inf.*, vol. 5, no. 2, pp. 147–156, May 2009.

[25] M. Metzger and G. Polakw, "A survey on applications of agent technology in industrial process control," *IEEE Trans. Ind. Inf.*, vol. 7, no. 4, pp. 570–581, Nov. 2011.

[26] A. Romanenko, L. O. Santos, and P. A. Afonso, "Application of agent technology concepts to the design of a fault-tolerant control system," *Control Eng. Practice*, vol. 15, no. 4, pp. 459–469, 2007.

[27] I. Seilonen, T. Pirttioja, and K. Koskinen, "Extending process automation systems with multi-agent techniques," *Eng. Applicat. Artif. Intell.*, vol. 22, no. 7, pp. 1056–1067, 2009.

[28] Y. S. Ng and R. Srinivasan, "Multi-agent based collaborative fault detection and identification in chemical processes," *Eng. Applicat. Artif. Intell.*, vol. 23, no. 6, pp. 934–949, 2010.

[29] C. H. Lo, Y. K. Wong, and A. B. Rad, "Intelligent system for process supervision and fault diagnosis in dynamic physical systems," *IEEE Trans. Ind. Electron.*, vol. 53, no. 2, pp. 581–592, Feb. 2006.

[30] C. H. Lo, E. H. K. Fung, and Y. K. Wong, "Intelligent automatic fault detection for actuator failures in aircraft," *IEEE Trans. Ind. Inf.*, vol. 5, no. 1, pp. 50–55, Feb. 2009.

[31] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discrete Event Dynam. Syst.: Theory Applicat.*, vol. 10, pp. 33–86, 2000.

[32] L. Ferrarini, M. Allevi, and A. Ded, "A methodology for fault isolation and identification in automated equipments," in *Proc. IEEE INDIN*, 2011, pp. 157–162.

[33] L. Ferrarini, M. Allevi, and A. Ded, "A real-time algorithm for fault identification in machining centres," in *Proc. 18th IFAC World Congress*, 2011, pp. 5201–5206.

[34] F. Jammes and H. Smit, "Service-oriented paradigms in industrial automation," *IEEE Trans. Ind. Inf.*, vol. 1, no. 1, pp. 62–70, Feb. 2005.

[35] G. Cândido, A. W. Colombo, J. Barata, and F. Jammes, "Service-oriented infrastructure to support the deployment of evolvable production systems," *IEEE Trans. Ind. Inf.*, vol. 7, no. 4, pp. 759–767, Nov. 2011.

[36] J. Puttonen, A. Lobov, and J. M. Lastra, "Semantics-based composition of factory automation processes encapsulated by web services," *IEEE Trans. Ind. Inf.*, vol. PP, no. 99, pp. 1–1, 2012.

[37] T. Cucinotta, A. Mancina, G. F. Anastasi, G. Lipari, L. Mangeruca, R. Checcozzo, and F. Rusin, "A real-time service-oriented architecture for industrial automation," *IEEE Trans. Ind. Inf.*, vol. 5, no. 3, pp. 267–277, Aug. 2009.

[38] I. Estvez-Ayres, P. Basanta-Val, M. Garca-Valls, J. A. Fisteus, and L. Almeida, "QOS-aware real-time composition algorithms for service-based applications," *IEEE Trans. Ind. Inf.*, vol. 5, no. 3, pp. 278–288, Aug. 2009.

[39] T. Cucinotta, L. Palopoli, L. Abeni, D. Faggioli, and G. Lipari, "On the integration of application level and resource level QoS control for real-time applications," *IEEE Trans. Ind. Inf.*, vol. 6, no. 4, pp. 479–491, Nov. 2010.

[40] M. Garcia-Valls, I. Rodriguez-Lopez, and L. Fernandez-Villar, "iland: An enhanced middleware for real-time reconfiguration of service oriented distributed real-time systems," *IEEE Trans. Ind. Inf.*, vol. PP, no. 99, pp. 1–1, 2012.

[41] L. Bassi, C. Secchi, M. Bonfè, and C. Fantuzzi, "A sysML-based methodology for manufacturing machinery modeling and design," *IEEE Trans. Mechatron.*, vol. 16, no. 6, pp. 1049–1062, Dec. 2011.

[42] K. Thramboulidis, "Model-integrated mechatronics—Toward a new paradigm in the development of manufacturing systems," *IEEE Trans. Ind. Inf.*, vol. 1, no. 1, pp. 54–61, Feb. 2005.

[43] C. Secchi, M. Bonfè, and C. Fantuzzi, "On the use of UML for modeling mechatronic systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 1, pp. 105–113, 2007.

[44] M. Witsch and B. Vogel-Heuser, "Towards a formal specification framework for manufacturing execution systems," *IEEE Trans. Ind. Inf.*, vol. 8, no. 2, pp. 311–320, May 2012.

[45] E. Estevez and M. Marcos, "Model-based validation of industrial control systems," *IEEE Trans. Ind. Inf.*, vol. 8, no. 2, pp. 302–310, May 2012.

[46] E. Estvez, M. Marcos, and D. Orive, "Automatic generation of PLC automation projects from component-based models," *Int. J. Adv. Manuf. Technol.*, vol. 35, no. 5–6, pp. 527–540, 2007.

[47] C. Fantuzzi, F. Fanfoni, C. Secchi, and M. Bonfè, "A design pattern for translating UML software models into IEC 61131-3 programming languages," in *Proc. 18th IFAC World Congr.*, 2011, pp. 9158–9163.

[48] V. Vyatkin, H.-M. Hanisch, C. Pang, and C.-H. Yang, “Closed-loop modeling in future automation system engineering and validation,” *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 39, no. 1, pp. 17–28, Jan. 2009.

[49] G. Cengic and K. Akesson, “On formal analysis of IEC 61499 applications, part a: Modeling,” *IEEE Trans. Ind. Inf.*, vol. 6, no. 2, pp. 136–144, May 2010.

[50] T. Strasser, A. Zoitl, J. H. Christensen, and C. Sünder, “Design and execution issues in IEC 61499 distributed automation and control systems,” *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 41, no. 1, pp. 41–51, Jan. 2011.

[51] K. Andersson, B. Lennartson, and M. Fabian, “Restarting manufacturing systems; restart states and restartability,” *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 486–499, Jul. 2010.

[52] K. Andersson, B. Lennartson, P. Falkman, and M. Fabian, “Generation of restart states for manufacturing cell controllers,” *Control Eng. Practice*, vol. 19, no. 9, pp. 1014–1022, 2011.

[53] G. Acampora, J. M. Cadenas, V. Loia, and E. M. Ballester, “Achieving memetic adaptability by means of agent-based machine learning,” *IEEE Trans. Ind. Inf.*, vol. 7, no. 4, pp. 557–569, Nov. 2011.

[54] D. Wang, “Robust data-driven modeling approach for real-time final product quality prediction in batch process operation,” *IEEE Trans. Ind. Inf.*, vol. 7, no. 2, pp. 371–377, May 2011.

[55] A. K. Pani and H. K. Mohanta, “A survey of data treatment techniques for soft sensor design,” *Chemical Product Process Modeling*, vol. 6, no. 1, 2011.

[56] D. Wang, J. Liu, and R. Srinivasan, “Data-driven soft sensor approach for quality prediction in a refining process,” *IEEE Trans. Ind. Inf.*, vol. 6, no. 1, pp. 11–17, Feb. 2010.

[57] L. Fortuna, S. Graziani, and M. G. Xibilia, “Comparison of soft-sensor design methods for industrial plants using small data sets,” *IEEE Trans. Instrum. Meas.*, vol. 58, no. 8, pp. 2444–2451, Aug. 2009.

[58] A. K. Pani and H. K. Mohanta, “Application of soft sensors in process monitoring and control: A review,” *IUP J. of Sci. Technol.*, vol. 5, no. 4, pp. 7–20, 2009.

[59] K. Y. Chan, S. Khadem, T. Dillon, V. Palade, J. Singh, and E. Chang, “Selection of significant on-road sensor data for short-term traffic flow forecasting using the Taguchi method,” *IEEE Trans. Ind. Inf.*, vol. 8, no. 2, pp. 255–266, May 2012.

[60] R. Luo and C.-C. Chang, “Multisensor fusion and integration: A review on approaches and its applications in mechatronics,” *IEEE Trans. Ind. Inf.*, vol. 8, no. 1, pp. 49–60, Feb. 2012.

[61] C. Caione, D. Brunelli, and L. Benini, “Distributed compressive sampling for lifetime optimization in dense wireless sensor networks,” *IEEE Trans. Ind. Inf.*, vol. 8, no. 1, pp. 30–40, Feb. 2012.

[62] V. Dubinin and V. Vyatkin, “Semantics-robust design patterns for IEC 61499,” *IEEE Trans. Ind. Inf.*, vol. 8, no. 2, pp. 279–290, May 2012.

Daniel Schütz received the degree in mechatronics from the University of Kassel, Kassel, Germany, in 2009. He is currently working toward the Ph.D. degree at the Institute of Automation and Information Systems, Technische Universität München, Garching, Germany.

His research focuses on the design of model-based development of intelligent real-time software for automation systems with regard to plant dependability and modularity aspects.

Andreas Wannagat received the degree in electrical engineering from the University of Wuppertal, Wuppertal, Germany, in 2004, and the Ph.D. degree from the Institute of Automation and Information Systems, Technische Universität München, Garching, Germany, in 2010.

His research areas are the development of software agents for real-time automation software and the dynamic reconfiguration of system components during runtime.

Christoph Legat received the degree in computer science from the Ludwig Maximilian University of Munich, Munich, Germany, in 2009. He is currently working toward the Ph.D. degree at the Institute of Automation and Information Systems, Technische Universität München, Garching, Germany.

He is member of the research group for knowledge modelling and retrieval of Siemens Corporate Technology, Munich, Germany. His current research interest focuses on the application of artificial intelligence techniques for engineering and operation of field-level automation software.

Birgit Vogel-Heuser (M’04–SM’12) received the degree in electrical engineering and Ph.D. degree in mechanical engineering from the RWTH Aachen, Aachen, Germany, in 1991.

She worked for nearly ten years in industrial automation in the machine and plant manufacturing industry. After holding different chairs of automation, since 2009, she has been head of the Institute of Automation and Information Systems, Technische Universität München, Garching, Germany. Her research work is focused on modeling and education in automation engineering for distributed and intelligent systems.