Validating Process Models in Systems Engineering Environments

Wikan Danar Sunindyo, Stefan Biffl
Christian Doppler Laboratory for Software Engineering Integration for Flexible Automation Systems
Vienna University of Technology
Vienna, Austria
e-mail: {wikan.sunindyo, stefan.biffl}@tuwien.ac.at

***Abstract*—Systems engineering environments usually involve multi-disciplinary engineering domains, such as mechanical, electrical, and software engineering, to cooperate and reach the common goals, e.g., to produce good quality products in time. Validation of systems engineering processes is challenging, because typically there are large numbers of possible combinations of heterogeneous sub-processes to test. In this paper, we propose a simulation-based approach to validate systems engineering processes. Simulation-based process validation (SbPV) aims at improving the efficiency of process validation in the real world. We collect engineering process data from the simulation of the systems engineering environments and analyze the conformance with the designed process model. As a real-world use case, we describe a production automation system and use a prototype implementation of the proposed SbPV approach to link the real engineering process information with the designed process model. Major result was the SbPV approach improved the efficiency of validating systems engineering processes and can complement testing of the sub-processes.**

Keywords: production automation systems, systems engineering processes, process validation, simulation.

I. INTRODUCTION

In modern software-based systems, such as production automation systems, the cooperation of engineering fields, e.g., mechanical, electrical, and software engineering, is required [1]. This kind of cooperation is called systems engineering, as software engineering provides an increasing share of added value to the resulting software-intensive systems and also depends on the seamless collaboration with all other engineering disciplines.

Systems engineering typically implements a large number of heterogeneous engineering processes that originate from different engineering disciplines, e.g., from mechanical, electrical and software engineering, and which have to be tested individually and also as a whole system. The testing of system-level systems engineering processes is not just aggregated testing all engineering sub-processes. Hence, unit testing of each sub-process is not sufficient for the validation of the whole system. Unit testing only deals with one sub-process, while we need to deal with large numbers of possible combinations of heterogeneous sub-processes to test. Hence, an advanced testing approach on system level is required to validate the numerous combinations of sub-processes that could possibly occur.

A production automation system is a typical example of systems engineering. Stakeholders of production automation systems consist of experts from different backgrounds, such as business management, electrical engineering, and software engineering. To reach their common goals, each stakeholder has to share his/her expert knowledge that is usually embedded in their different tools and data models. However, the information needed by a stakeholder from other stakeholders usually depends on the specific goals and tasks. For example, the business manager may want to know how many products will get produced regarding certain business orders assigned for a certain period, while key parts of this information are available only on the run-time level of the production process, e.g., numbers of finished products and numbers of unfinished products. Process analysis based on run-time events can provide valuable input to discover the kind of information needed by the business manager: a timely view on the actual conditions of the system, which can be compared with the designed model for process conformance.

In this paper, we propose simulation-based process validation (SbPV) [2] as an alternative to validate processes on system level. Currently process validation on system level is done by e.g. checking manually whether the processes are correctly done to produce products required by the business manager. SbPV is focusing more on validating processes rather than testing of products. SbPV is used to simulate the main behavior of the system and test different kinds of parameters to validate the system, e.g., by measuring the number of finished products. Major expected benefits of SbPV are that (a) simulation allows (automated) process validation on various levels, such as between the business level and the process level and between the machine level and the process level; (b) simulation is the foundation for capturing data on process level with respect to run-time data [3]; and (c) when comparing runtime data with simulation data, this can improve the simulation and support runtime diagnosis.

We simulate production automation system engineering processes describing how to design and build machine configurations based on inputs of business managers. The used software simulation tool is called Simulator of Assembly Workshops (SAW) [4, 5]. Engineering processes[6] in the SAW system are defined by inputs, outputs, and activities to process the inputs to get transformed into the outputs, e.g., by using a set of business orders as inputs, we assemble a new configuration of machine layouts as an output using specific sets of machine functions. A machine function is a device that builds a new product from some raw materials/other products. One machine can include some machine functions.

SAW simulates the components of real manufacturing systems and their behavior based on multi-agent technology validated with the behavior of the equivalent lab hardware[1]. The simulator accepts business orders, dispatches work orders, and schedules these work orders to assemble products according to the business orders. During the production processes, the simulator collects event logs of each activity, e.g., starting and finishing a product. With a process analysis tool, such as ProM[2] [7], researchers and practitioners can analyze these process event logs for conformance checking between the designed process model and actual running processes.

Major result of this work is that the SbPV approach can show the conformance of actual process models with expected process models. The SbPV approach can be applied and generalized to other engineering systems, e.g., power plant systems to discover hidden information from the engineering process at run time, for example the organizational structure of the engineering processes.

The remainder of this paper is structured as follows: Section 2 summarizes related work on production automation systems, SbPV, and process analysis. Section 3 motivates the research issues. Section 4 develops the solution approach to enable the event-based analysis of production automation system engineering processes. Section 5 describes the evaluation results. Section 6 discusses lessons learned and concludes the paper.

II. Related Work

This section summarizes related work on production automation systems, simulation-based process validation (SbPV), and process analysis and improvement.

A. *Production Automation Engineering*

Industrial production automation systems include manufacturing systems, such as assembly workshops that combine smaller parts into more complex products, e.g., cars or furniture. Several domains have to cooperate for manufacturing: (a) business order processing and work order scheduling, (b) technical processes for workshop and systems coordination, and (c) technical designs of a set of machines in a defined workshop layout [8, 9].

In typical engineering disciplines, models (e.g., model-based design and testing [10]) help to construct new systems products and to verify and validate the solutions regarding the requirements, specification, and design models. Traditional systems engineering processes follow a water-fall like engineering process with late testing approaches [11]. Unfortunately, insufficient attention is paid in the field of automated systems engineering to capabilities for Quality Assurance (QA) of software-relevant artifacts and change management across engineering domains [12], possibly due to technical and semantic gaps in the engineering team. Thus, there is considerably higher effort for testing and repair, if defects get identified late in the engineering process.

B. *Simulation-based Process Validation*

Hass [6] defined testing as a process that can be described and hence monitored and improved. She explained the purposes of the test process to provide information to assure the quality of the product, decisions, and the processes for a testing assignment. This approach is useful to conduct unit testing. However, other types of process validation, like Simulation-based Process Validation (SbPV), have to complement unit testing.

There are several research reports on SbPV for complex systems. Sargent [13] discussed the verification and validation of simulation models. He uses simulation to model the real world. Hence, verification and validation needs to be done to check the conformance between the simulation world and the real world, e.g., theory validation, conceptual model validation, specification verification, implementation verification, and operational (results) validation. However, the validation of relationships between components of the systems that could affect the behavior of the systems is not discussed in detail.

Carpanzano and Ballarino [2] proposed a structured approach for designing and implementing SbPV for factory automation systems. They used a closed-loop SbPV method to verify the different hierarchic levels of the designed automation system in a modular way. They simplified the description of the complex system and enhanced model reuse through the adoption of formal models that exploit the concepts of modularity, encapsulation, and abstraction. SbPV allows verifying the correctness the method before final implementation and reliability is improved. However, this approach is still focusing on using static information for testing the simulation and does not investigate in sufficient detail the knowledge management of different components from various stakeholders of the systems.

C. *Process Analysis and Improvement*

The usage of process modeling approaches for measuring and analyzing the conformance between designed and actual process models has been applied for different domains. Process analysis has been applied to complex systems, like workflow management systems, Enterprise Resource Planning or Customer Relationship Management systems. Van der Aalst et al. [7] use workflow technologies to structure the processes running inside IT systems. This workflow technology supports event provision that could be useful for process analysis in software engineering by enabling particular models that link basic tool events to process events. This approach is called process mining and uses stored events, which refer to tasks and process cases originating from people/tools/systems, to monitor and analyze real workflows with respect to designed workflows. Exemplary application scenarios are process discovery, performance analysis, and conformance checking. Process mining has been implemented in the open source tool ProM

[1] http://www.acin.tuwien.ac.at

[2] http://www.processmining.org

and can be used to discover the process model based on the available event log, analyze the performance of the processes and suggest possible process improvement candidates.

Gerke et al. [14] propose to use a process modeling approach for solving problems of reference models. A reference model provides a set of generally accepted best practices to create efficient processes to be deployed inside organizations. The challenge in a reference model is to determine how these best practices are implemented in practice. The authors propose a new approach and algorithm which allow measuring the compliance of process models with reference models [14]. However, this approach focuses more on the building of process models rather than on the relationships between components of the model that can illustrate further the interaction inside an organization.

Rembert and Ellis [15] extended process mining techniques, which focused on mining the control-flow of business processes, towards analyzing multiple perspectives of a business process. The extension of the process mining techniques includes explaining formal and general definitions of a business process perspective and presenting the approach to mine other business process perspectives using these definitions, i.e., the behavioral perspective and the role assignment perspective, that can be useful for analyzing processes in the SE context.

III. RESEARCH ISSUES

The scope of this research is the investigation of simulation-based process validation (SbPV) approach in systems engineering environments, especially business processes in the production automation systems. The business processes of production automation systems involve cooperation between different engineering stakeholders, e.g., mechanical, electrical, and software engineering.

The quality managers of the production automation systems need to be able to check the status of business orders, e.g., whether the business orders can be fulfilled or not. To do this, the quality managers should collect and analyze data from different stakeholders, e.g., from business manager, workshop configurator and workshop operator, to obtain required information that will be useful to make decision on the status of business orders.

However, collecting and analyzing data in the production automation systems are difficult task, because (1) heterogeneous data formats and tools used by different stakeholders make it hard to integrate the data, (2) limited analysis approaches to analyze production automation data, (3) limited validation approaches on analysis results of production automation data, (4) limited evaluation on validation approaches so far.

In this paper, we propose a simulation based process validation approach to solve the current limitations on collecting and analyzing production automation systems data. We derive detail research issues as follows.

RI-1. How can we validate the business goal in the production automation systems? The business goal of production automation systems need to be validated by the quality managers, as a way to discover the status of business orders. In this case, the quality managers collect data from two different stakeholders layers, namely from business layer and from process layer. In the business layer, the goal is to provide certain number products in some periods, which can be achieved by setting some parameters, e.g., the number of pallets, the number of orders, and the types of products. In the process layer, the goal is to build a right layout configuration of machines that can support production, by setting the configuration parameters, e.g., number of machines, type of machine functions, and direction of conveyors. The quality managers validate the business goal in the business level and the production processes in the process level, by integrating the information from both levels. We propose a software architecture on business processes to validate the business goal in the production automation systems.

RI-2. How can we evaluate business processes in the production automation systems? To reach the business goal, business layer and process layer in the production automation systems will produce process information that can be useful for status monitoring by the quality managers. However, the link of process information between business layer and process layer is not straightforward due to heterogeneous formats and data types used by both layers. In the business layer, the process information is in the form of product trees, which contain information of product parts and machine function needed to build products. In the process layer, the process information is in the form of process events, which contain information of all events during production processes. We propose an approach to evaluate the business processes in the production automation systems by checking conformance between the product trees and the process event.

IV. SOLUTION APPROACHES

This section explains the solution approaches used to solve the research issues in section 3. The solutions are including validating the business goal and evaluating the business processes in the production automation systems.

A. Business Goal Validation

To validate the business goal in the production automation systems, we propose a software architecture that consists of different layers of production automation components and show how the components interact with each other.

Figure 1 illustrates the business goal validation architecture which consists of three layers, namely business layer, process layer, and machine layer. In the business layer, the business manager dispatches and schedules business orders, which contain specification of products in the form of product trees and other required information, e.g., the number of products and due date of production process.

In the process layer, the workshop configurator configures the layout of simulation of assembly workshop (SAW) [5] that supports the business manager's requirements. The workshop operator runs the simulation based on the business orders. Each process event during the production process is stored in the process event log for further purposes, e.g., process analysis by quality manager.

In the middle of process layer, there is a schematic view of an assembly workshop, which consists of heterogeneous agent-controlled components, namely 40 conveyors, 17 junctions, more than 15 pallets, 3 product parts and storage areas, 6 machines and 4 robots.

In the machine layer, the system engineer provides the real workshop systems that interact with real machines which are provided by the machine vendor. In this paper, we don't discuss further about the machine layer and put focus on the interaction between the business layer and the process layer.

The business goal is validated by making simulation on how the assembly workshop runs and produces products which are stated in the business orders. The simulation is done by SAW simulator, which runs different test cases to validate different workshop configurations and different business orders. During production process simulation, the SAW simulator produces process events e.g., starting events, finishing events, and other relevant events which are stored in the process event log. We can set the relevant parameters of simulation, e.g., failure classes, scheduling strategy, and number of pallets, to accommodate different risks and situations that could be faced in the real assembly workshop.

The results of the simulation can be used as input for real assembly workshop in the machine layer. The usage of a software simulator here is needed to accommodate the reconfiguration of the production automation system in order to get a better performance. Validation on hardware test bed is expensive, hence we build software simulator with agent-controlled components that imitate behaviors of real components in the real system.

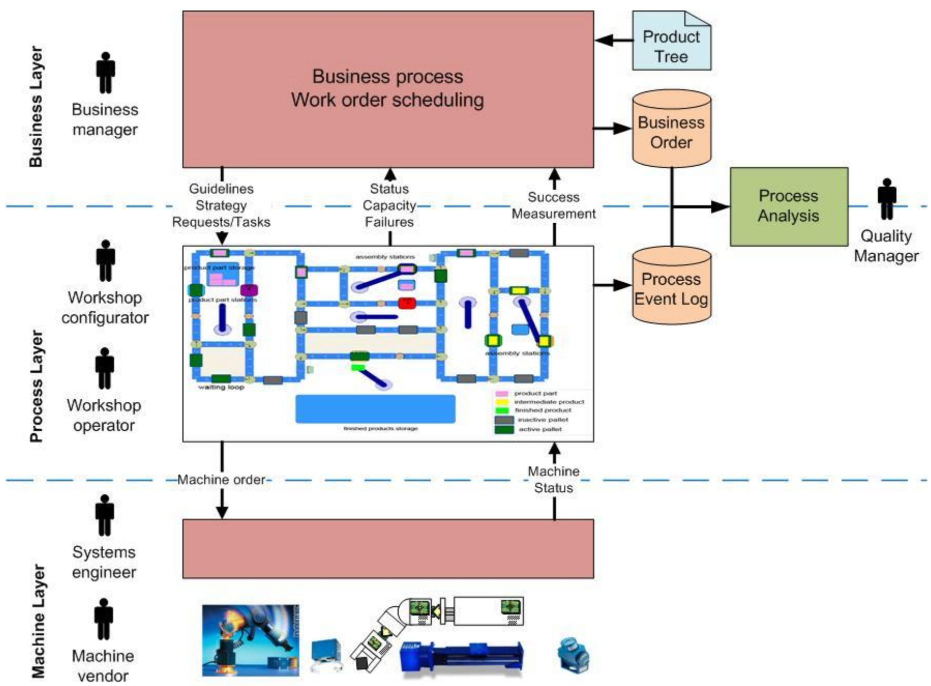

Figure 1. Business Goal ValidationArchitecture for Production Automation Systems.

For test cases on the business goal validation, we design experiments on the simulator by setting several parameters on the business orders fed into the simulator. The business orders consist of 1,500 products with two fictional types of products, named Billy Medium and Billy Complex. Billy Medium consists of one simple part and one intermediate part, while Billy Complex consists of two intermediate parts. We evaluate 40 test cases of business goals, by comparing the results of simulating 4 different classes of failures.

The classification of classes of failures is based on the risk of machine failures and/or conveyor failures in the simulation workshop, according to the position and the importance of the machine and conveyor for the overall system (refer to Table 1). For effective comparison of the robustness of workflow scheduling strategies regarding their exposure to failures in the transportation system, we used First Come First Served (FCFS) strategy, which execute the first allocated task first.

The explanation of test cases with different class of failures is as follows. (1) C0 consists of test cases with no failure. (2) C1 consists of test cases with 5 conveyor failures in each test case. (3) C2 consists of test cases with 2 machine failures in each test case. (4) C3 consists of test cases with combination of 5 conveyor failures and 2 machine failures in each test case. The machine failures and the conveyor failures occur randomly in the test cases. The result of business process validation is discussed in section 5.

TABLE I. FAILURE CLASSES AND RISK ANALYSIS

Classes of Failure	Failure Impact
C0	No Failure
C1	Conveyor Failures
C2	Machine Failures
C3	Combined Conveyor and Machine Failures

B. *Business Processes Evaluation*

To evaluate the business processes in the production automation systems, we collect and analyze the process event data from running experiment on the SAW simulator. The process event logs contain activities of each component of the SAW simulator in the form of XML files. These files consist of attributes that explain the identifier for test run, identifier of event, timestamp, type of event, identifier of order, identifier of work piece, and component name. The detail information about the experiment can be found in section 4.1. The evaluation of business processes will be done by conformance checking between the designed process model (product trees from the business layer) and the generated process model from actual data (process event log from the process layer).

For analyzing the process event data, we use Process Mining (ProM) tool to generate process model from process event logs. ProM is an open-source tool for implementing process and organizational mining techniques in a standard environment, which allows the extraction of information from event logs. The using of ProM is based on the minimal amount of information that needs to be present in the general cooperative information systems. The event log should follow these requirements i.e., each logged event should be a single event that occurred at a defined point in time, each logged event should refer to one single activity only, each logged event should contain a description of the event that happened with respect to the activity, each logged event should refer to a specific process instance (case), and each process instance should belong to a specific process. The originator of the event is optional information for the event. This information is useful for advanced analysis, i.e., organizational mining.

To generate the process model from process event logs, we should transform the format of SAW event logs into mining XML (format of ProM files). **Fehler! Verweisquelle konnte nicht gefunden werden.** shows the SAW event logs (top) and its transformation (bottom). The transformation is as follows. (1) The OrderId becomes ProcessInstance Id. (2) The type of event becomes EventType. (3) The Workpiece Id becomes Workflow Model Element. (4) The Component Name becomes Originator. (5) The Timestamp is calculated to get the date and time format. We use this transformed file as an input for ProM tool and produce a process model for conformance checking with the product trees in the business layer.

V. RESULTS

In this section, we describe the results of our solution approaches, namely the business goal validation and the business processes evaluation in the context of production automation systems.

A. *Business Goal Validation*

The validation of business goal is done by running several experiments on the SAW simulator with different parameters, e.g., classes of failures and the product types. Figure 3 shows our experiment results and illustrates the relationships between failure classes and system outputs for different product types, i.e., Billy Medium and Billy Complex. A system output is defined as a number of finished products, and is correlated positively with the class of failures and the product types.

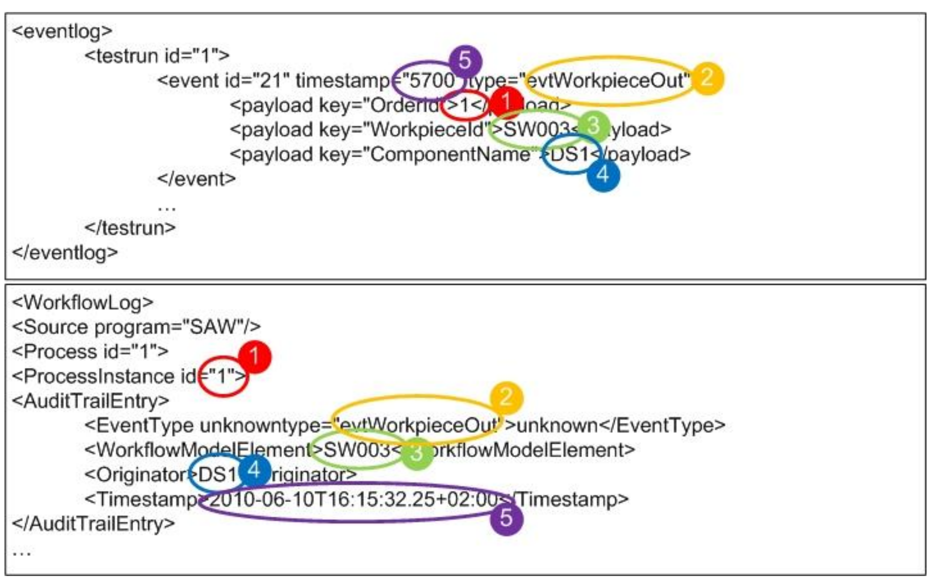

Figure 2. Structure and Transformation of SAW event logs.

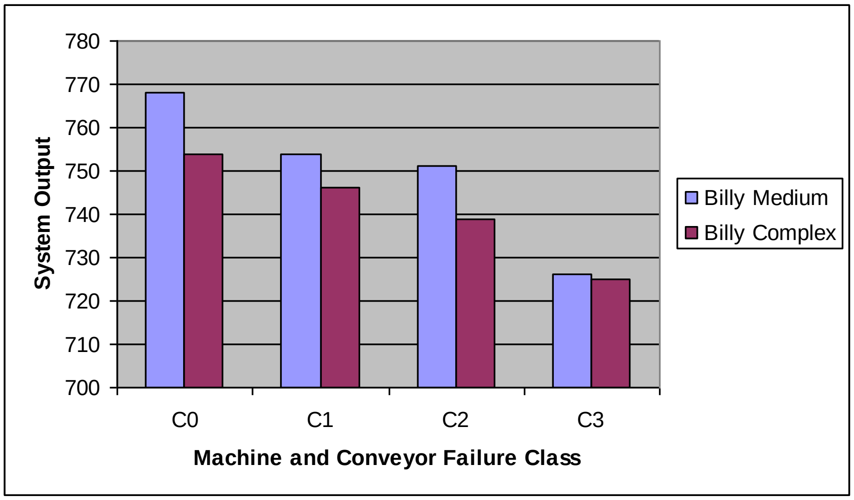

Figure 3. Relationships between failure classes and system output-

The more severe the failure is, the fewer products are finished, i.e., the combination of machine and conveyor failures produces the fewest number of products. The production of Billy Complex is more difficult than the production of Billy Medium, because it has more input raw materials and has more processing steps and machine functions used than the Billy Medium. Hence the number of Billy Complex products finished is lower than the number of Billy Medium products finished. This information is useful in reconfiguring the real machine layout in the mechanical engineering domain. The possibility of machine failures and conveyor failures in the real machine layout is one of our considerations to configure machine with similar functions in separate routes. Other considerations could be the type and the number of products, the number of pallets, and the storage of products.

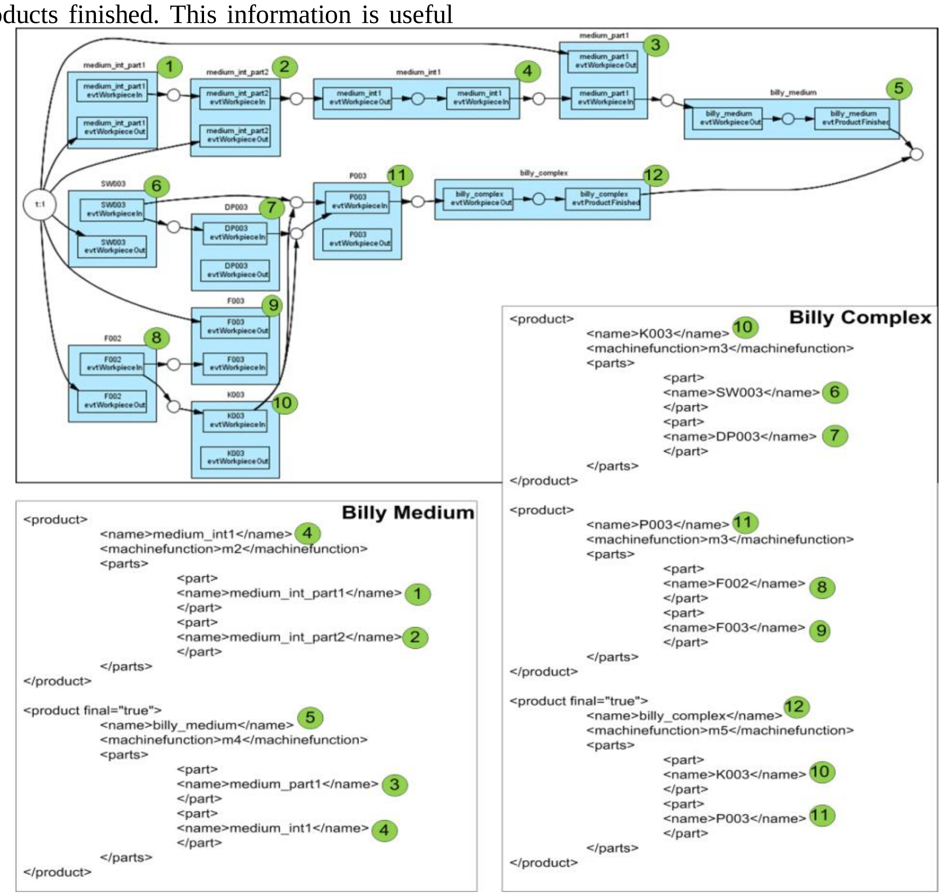

Figure 4. Overview Process Model and Product Tree Conformance.

B. Business Processes Evaluation

The evaluation of business processes in the production automation systems is done by conformance checking between the designed process model in the business layer and the generated process model from actual process event data in the process layer. The top of **Fehler! Verweisquelle konnte nicht gefunden werden.** shows the process model from actual process event data, which is generated by using Alpha Algorithm plugin of ProM. This process model conforms to the product trees (see bottom of **Fehler! Verweisquelle konnte nicht gefunden werden.**) of two product types, namely *Billy Medium* and *Billy Complex*. A product tree consists of description of products, the parts to build the product and machine function that build the product from its parts. The product tree is written in XML notation and can be illustrated as a tree with the product as a root and its parts as nodes and a machine connects between the product and its parts.

The process model is generated by analyzing event logs using the process analysis tool ProM. The model shows the structure of products building from its part. Billy Medium product (5) is built from medium_part1 (3) and medium_int1 (4). The medium_int1 (4) is built from two raw materials, namely medium_int_part1 (1) and medium_int_part2 (2). Billy Complex product (12) is built from two intermediate materials, namely K003 (10) and P003 (11). P003 (11) is built from F002 (8) and F003 (9), while K003 (10) is built from SW003 (6) and DP003 (7). The way how to arrange this product in the run time can be shown by using the process model illustrating that the materials in the process model are matching with the materials from the product trees with the same numbers. In this case, we have checked conformance between the actual process model and the product trees.

VI. DISCUSSION AND CONCLUSION

In this paper, we have used the SbPV approach to improve the efficiency of process validation in the systems engineering domain, by using process validation of the simulation instead of complete process validation of real machine layouts. We used production automation systems as an example of systems engineering that involve engineering processes from different engineering disciplines, like mechanical, electrical, and software engineering. Simulation of production automation systems is implemented in the form of SAW software simulator, which supports the reconfiguration of real machine layouts based on inputs and requirements of business managers.

Previous research [5] on testing and improving production automation systems has been performed by doing performance evaluation of workflow scheduling strategies using different kinds of parameters, e.g., transportation times and conveyor failures. However, the approach was not considering the structure and relationships changes between machines and components during run time. Other research proposed on using of ontology areas to bridge semantic gaps between stakeholders in the production automation domain [16]. However, this approach was not considering the maintenance efforts of system components.

In the SbPV approach, we implement a process analysis approach to explore the information from event logs collected during the simulator run time. By using process analysis, we can derive process models illustrating the process flow during run time. The deviations between the designed process model or product trees and the actual process model can be seen and detected by using such process analysis approaches.

Business Goal Validation. From the results section, we see that by using the SbPV approach, we can validate the business goal by simulating the real assembly workshop with a software simulation that provides possibility to change different parameters of configuration. The results show that there is a positive correlation between the system outputs and the classes of failures. The more severe the failure, the lesser numbers of products finished.

Business Processes Evaluation. By using conformance checking approach, we can evaluate the business processes in the process layers to the product trees in the business layer. This approach can support the quality managers to inspect whether any deviation happens during the simulation running, by checking the generated process model with the product trees. However, the current evaluation is more focused on machines structures and their relationships, while interactions with other components of the systems, e.g., conveyors and junctions are not discussed yet.

Future Work. Future work will be to analyze challenging engineering processes and environments by using organizational mining approaches or other advanced approaches. Furthermore, we plan to use more complex simulation environments with more machines and more complex routes, investigate them and then compare with results of the presented research.

ACKNOWLEDGMENT

This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria. This work has been partially funded by the Vienna University of Technology, in the Complex Systems Design and Engineering Lab. The authors also want to acknowledge the inputs and discussion with Thomas Moser and Dietmar Winkler.

REFERENCES

[1] S. Biffl and A. Schatten, "A Platform for Service-Oriented Integration of Software Engineering Environments," presented at the Eight Conference on New Trends in Software Methodologies, Tools and Techniques (SoMeT 09), 2009.

[2] E. Carpanzano and A. Ballarino, "A structured approach to the design and simulation-based testing of factory automation systems," in IEEE International Symposium on Industrial Electronics (ISIE 2002), 2002, pp. 181-186.

[3] I. Terzic, M. Merdan, A. Zoitl, and I. Hegny, "Modular assembly machine - ontology based concept," in IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008), 2008, pp. 241-244.

[4] P. Vrba, "MAST: Manufacturing Agent Simulation Tool," presented at the Emerging Technologies and Factory Automation, 2003.

[5] M. Merdan, T. Moser, D. Wahyudin, and S. Biffl, "Performance evaluation of workflow scheduling strategies considering transportation times and conveyor failures," in IEEE International Conference on Industrial Engineering and Engineering Management (IEEM 2008), 2008, pp. 389-394.

[6] A. M. J. Hass, "Testing Processes," in IEEE International Conference on Software Testing Verification and Validation Workshop, 2008 (ICSTW '08), 2008, pp. 321-327.

[7] W. M. P. van der Aalst, A. J. M. M. Weijters, and L. Maruster., "Workflow Mining: Discovering Process Models from Event Logs," IEEE Transactions on Knowledge and Data Engineering, vol. 16, pp. 1128-1142, 2004.

[8] D. Winkler, S. Biffl, and T. Östreicher, "Test-Driven Automation - Adopting Test-First Development to Improve Automation Systems Engineering Processes," presented at the 16th EuroSPI Conference, Madrid, Spain, 2009.

[9] A. Lüder, J. Peschke, and D. Reinelt, "Possibilities and Limitations of the Application of Agent Systems in Control," presented at the International Conference On Concurrent Enterprising (ICE), 2006.

[10] P. Baker, R. D. Zhen, J. Gabrowksi, and H. Oystein, Model-Driven Testing: Using the UML Testing Profile: Springer, 2008.

[11] GAMP 5, "Good Automated Manufacturing Practice," International Society for Pharmaceutical Engineering (ISPE), 2008.

[12] W. Schäfer and H. Wehrheim, "The Challenges of Building Advanced Mechatronic Systems," in 2007 Future of Software Engineering - International Conference on Software Engineering, Washington, DC, 2007, pp. 72-84.

[13] R. G. Sargent, "Verification and validation of simulation models," presented at the Proceedings of the 40th Conference on Winter Simulation, Miami, Florida, 2008.

[14] K. Gerke, J. Cardoso, and A. Claus, "Measuring the Compliance of Processes with Reference Models," in On the Move to Meaningful Internet Systems: OTM 2009, ed, 2009, pp. 76-93.

[15] A. J. Rembert and C. Ellis, "An initial approach to mining multiple perspectives of a business process," presented at the 5th Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations, Portland, Oregon, 2009.

[16] T. Moser, S. Biffl, W. Sunindyo, and D. Winkler, "Integrating Production Automation Expert Knowledge Across Engineering Stakeholder Domains," presented at the 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2010), Krakow, Poland, 2010.