Process Analysis and Organizational Mining in Production Automation Systems Engineering

Wikan Danar Sunindyo Thomas Moser Dietmar Winkler Stefan Biffl
Christian Doppler Laboratory for
Software Engineering Integration for Flexible Automation Systems
Vienna University of Technology, Austria
{wikan, moser, winkler, biffl} @ifs.tuwien.ac.at

Abstract. The engineering and operation of cooperative information systems, such as production automation systems, typically involves stakeholders from several backgrounds, e.g., business, electrical and software engineering. Heterogeneous stakeholders of production automation systems, such as both business and technical stakeholders, need to cooperate and share information to perform and monitor production processes. Major challenges are to validate the run-time system processes with the designed processes and to analyze the structure of the components of the systems for analysis and decision making. In this paper, we investigate an approach for process analysis and organizational mining to support monitoring and improvement of systems processes based on integrated events logged during run time on simulation of production automation systems. Major result in the study context was that the proposed process analysis and organizational mining approach provided a better view on the structure and relationships of the system components and improved the system production output.

Keywords: business process management and integration; process modeling, analysis and design.

1 Introduction

Production automation systems as cases of cooperative information systems involve different layers of stakeholders that need to cooperate to reach their common goals. Stakeholders of the systems consist of experts from different backgrounds, such as business, electrical and software engineering. To reach their common goals, each stakeholder should share their 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 much products he will get produced regarding certain business orders he assigned for a certain period, while key parts of this information are available only on the run-time level of the production process. Process analysis based on run-time events can provide valuable input to discover the kind of information needed by the business manager. The analysis of run-time events can give a timely view on the actual conditions of the system and can be compared with the designed model for conformance.

In this paper, we propose approaches for process analysis and organizational mining in the production automation engineering domain. The process analysis approach can help the business/project manager to check the conformance of the run-time processes with the designed process model. The organizational mining of events and processes can help the project manager to investigate the structure and interactions between the systems components during run time. These analysis results can provide significant input to maintenance planning and further decision making. We evaluate the proposed approach with data from a real-world production use case.

As a running use case from the production automation domain we use the assembly of complex products from simpler product parts, implemented as the Simulator of Assembly Workshops (SAW). This simulator simulates components of real manufacturing systems and their behavior based on multi-agent systems 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 on each activity, like starting and finishing a product. With a process analysis tool like ProM[2] [20], researchers and practitioners can analyze the process event log to perform conformance checking between the designed process model and actual running processes.

Major results of this work are the process analysis and organizational mining approaches show certain relationships between different parameters of the SAW simulator, e.g., classes of failures and system outputs, and the relationships of the similar machines and the machines which are working together during run time. These approaches can be applied and generalized to other engineering systems to discover hidden information from the engineering process in the run time. By using information from run time, we can analyze model process and structure of the systems from different views, rather than just using designed static models.

The remainder of this paper is structured as follows: Section 2 summarizes related work on production automation system, semantic integration technologies, and process analysis. Section 3 identifies the research issues and the research method. Section 4 develops the solution approach to enable the event-based analysis of production automation system processes. Section 5 describes the empirical study results. Section 6 discusses lessons learned and concludes the paper.

2 Related Work

This section summarizes related work on engineering of production automation systems, semantic integration technologies, and on process analysis and organizational mining in collaborative systems.

---

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

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

2.1 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 [9, 24].

In engineering disciplines models (e.g., model-based design and testing [1]) 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 [4]. 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 [17], 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.

As production automation system for research the Manufacturing Agent Simulation Tool (MAST) [22] provides a unique combination of multi-agent-based manufacturing control features and a simulator of the manufacturing environment. The system can represent the full range of roles involved in such a complex system: business orders, workshop scheduling strategies, robust workshop control, and control of individual machines in the workshop. The Simulator of Assembly Workshops (SAW) [10] is an extension of the original MAST simulator providing means for automatic execution and evaluation of a predefined set of simulation experiments that can be used for comprehensive statistical data analysis.

A general challenge for process assessment is reliable and efficient data collection on the actual practices in a software development project as data collection mostly relies on human reporting and document analysis [23]. Despite positive findings in validation studies, there is still a gap between the published research results and the successful adoption of objective quality evaluation approaches in real-world projects due to the poor quality of available project data, heterogeneous data sources, and the effort required to extract and collect the necessary data [3, 8, 12]. Semantic integration approaches can help to improve data exchange and systematic tracing between models across these disciplines.

2.2 Semantic Integration Technologies

Semantic integration is defined as the solving of problems originating from the intent to share information across disparate and semantically heterogeneous data. These problems include the matching of data schemas, the detection of duplicate entries, the reconciliation of inconsistencies, and the modeling of complex relations in different sources [14]. In an engineering context common engineering concepts can be used as basis for mappings between proprietary tool-specific engineering knowledge and more generic domain-specific engineering knowledge to support transformation between these engineering tools. Engineering knowledge descriptions in ontologies can be the basis to provide mappings between these ontologies to enable semantic integration. Noy [13] identified three major dimensions of the application of ontologies for supporting semantic integration: the task of finding mappings (i.e., semantic correspondences) semi-automatically, the declarative formal representation of these mappings, and reasoning using these mappings. Semantic mappings can either consist of elements of the same granularity, such as 1:1 mappings of concepts or attributes, or in addition may be expressed using different levels of granularity, such as mapping the attribute of a concept to a target concept or mapping a concept to all inherited subconcepts of a target concept.

An ontology is a representation vocabulary for a specific domain or subject matter. The ontology does not seek to describe all the knowledge contained in a domain, but only the knowledge essential for conceptualizing the domain [6]. The use of ontologies for knowledge representation, sharing and high-level reasoning is a major topic in the area of designing flexible and autonomous systems, such as agent-based control systems [15]. Ontologies are necessary for enabling data-driven system configuration and re-configuration in an extensible manner. The use of domain ontologies for representing knowledge in the factory automation domain has been proposed [7].

Moser et al. [2] introduced the Engineering Knowledge Base (EKB) framework as a semantic web technology approach for addressing challenges coming from data heterogeneity that can be applied for a range domains, e.g., in the production automation domain [2] and also SE. Further, Biffl et al. [2] used the approach for solving similar problems in the context of Open Source Software projects, in particular, frequent-release software projects.

2.3 Process Analysis and Improvement

The use 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. use workflow technology to structure the processes running inside IT systems. This workflow technology supports events provision that could be useful for process analysis in SE by enabling particular models that link basic tool events to process/workflow events [20].

Gerke et al. [5] propose using 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 [5]. However this paper does not discuss about the relationships between components of the model that can illustrate the interaction inside an organization better.

Another approach was proposed by van der Aalst et al. [20]. This approach uses stored events, which refer to tasks and process cases coming from people/tools/-systems, to monitor and analyze real workflows with respect to designed workflows. This approach is called process mining, and can be used for process discovery, performance analysis, and conformance checking. The approach has been implemented in the open source tool ProM 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.

Rembert and Ellis [16] 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.

Song and van der Aalst improve their process analysis approach by adding organizational mining that enable people to discover organizational models and social networks in the information system [18]. These models can assist in improving the underlying processes. In this paper, the authors distinguish between three types of organizational mining: (1) organizational model mining, (2) social network analysis, and (3) information flows between organizational entities. Organizational mining can benefit from creatively using approaches developed for the process perspective. This approach can be used to illustrate the interaction between components of the system for other domains, including the production automation domain.

3 Research Issues

The scope of this research is the investigation of process analysis for cooperative information systems, for example industrial assembly workshops, which supports the cooperation of stakeholders from different layers, such as the business layer (what to produce when) and the workshop layer (how to organize the production process).

Experts from different layers have different requirements and challenges to answer. For example, business experts deal with managing business orders from customers and then feed these customer orders into the simulation workshop. One exemplary analysis they want to perform may be to check how many products are finished based on the given business orders. To do this, they should communicate with other experts from different layers, for example process experts.

Process experts manage the simulation components, set parameters for simulation and capture data such as the number of finished products by analyzing the event logs. They also need to react to machine failures and conveyor failures occurring in the simulation workshop, since these failures could disturb the production process. The goals of the evaluation are (a) to show the benefits of integrated event capturing across layers, (b) to illustrate the capabilities of the proposed event analysis processes, and (c) to analyze the process model and organization model of the process events.

In this paper, we use a process analysis approach based on heterogeneous event logs from different layers to perform a useful analysis for decision making of the SAW simulation system, namely descriptive statistics analyses, ProM [20] works with Petri nets for process modeling, and organization mining to reveal the structure and relationship between machines used in the simulator. Based on the use case from the automated manufacturing domain, we derive the following research issues:

RI-1. How to check conformance between designed process model and actual process model. Cooperative information systems typically consist of different layers and involve heterogeneous components. The question usually asked by the system stakeholders is how to check conformance between designed process model and actual process model. The SAW simulator, as a use case of cooperative information systems, consists of different layers of stakeholders. The designed process model is on the business layer, which consists of the parts, the machine functions, and the ways to build a product. This designed process model is called a product tree. The actual process model can be obtained by analyzing event log data collected during the simulation was ran. We perform a conformance checking between designed process model and actual process model.

RI-2. How to find the structure of components of the system during the run time. The heterogeneities of components and layers in the cooperative information systems and different scopes used in the systems, e.g., design time and run time, make the identification of structure and relationships of components difficult and time-consuming. For instance, we want to discover the relationships between machines that working together to finish a product and the relationships between machines that have similar functions, from the run time information. In the design time, we just know the information of each machine individually and are not connected by certain algorithms. By using analysis/organizational mining on event log collected during time, we can discover the relationships between machines. This information is useful for further decision, e.g., (1) counting how many times a machine is used so we can suggest maintenance intervals after the machine is running for a certain time; (2) identifying machines with similar functions so we can suggest the replacement of a broken machine with other machines with similar functions.

4 Study

This section introduces the process analysis and organizational mining approaches for the production automation engineering domain and describes its architecture. We identify a set of use cases form assembly workshop engineering, describe the use cases in the context of the SAW production automation research system [11] and show how the approaches can be applied to the use cases.

Figure 1 illustrates the use case scenario showing the process analysis approach for the SAW production automation system. In this figure, there are three layers of the production automation system, namely business layer, process layer, and machine layer. In the central of this figure, there is a schematic view on 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 business layer, the business manager dispatches and schedules business orders, which contain specification of products from product trees and other information required, for example the number of products and due date of production process.

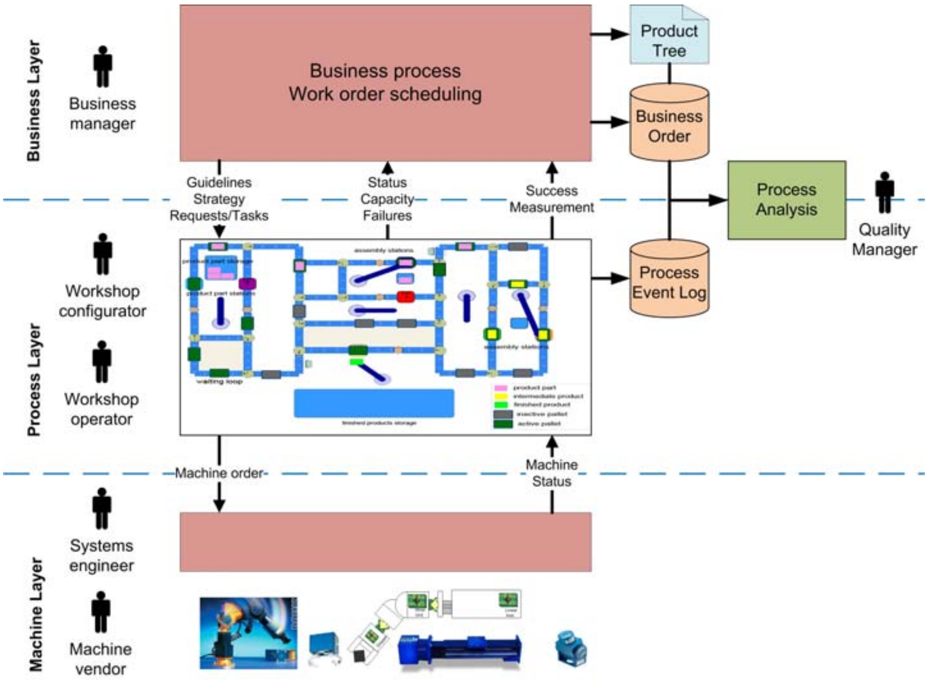

Figure 1. Architecture of process analysis in production automation domain.

This information is useful to build process simulation in process layer to simulate the workshop by setting all relevant parameters, like failure classes, scheduling strategy and number of pallets. The SAW simulator runs different test cases and produces process event log that contains starting events, finishing events, and other relevant events. The information from process event log together with the information from business order will be integrated and analyzed to get some analysis results about the correlation of information between different layers that can be used to improve processes and products.

The result from simulator can be used for input for the real machines 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.

In this paper, we design experiments on the simulator by setting several parameters on business orders fed into the simulator. Business orders consist of 1500 products, with two types of products, namely *Billy Medium* and *Billy Complex*. The product trees of *Billy Medium* and *Billy Complex* are illustrated in XML files in figure 5. *Billy Medium* consists of one simple part and one intermediate part, while *Billy Complex* consists of two intermediate parts. We evaluate 40 test cases to compare the results from different failure specifications.

In this paper, we design experiments on the simulator by setting several parameters on business orders fed into the simulator. Business orders consist of constant number of products with two types of products, namely billy complex and billy medium. The product trees of Billy Medium and Billy Complex are illustrated in XML files in figure 4. We evaluate numbers of test cases to compare the results with different failure specifications.

The failure specification consists of the identifier of the affected resource to fail, the start and end points in time of the occurrence of the failure. We classified the risk of a failing machine and a failing conveyor for all machines and conveyors in the workshop in 4 failure classes, according to the position and the importance of the machine and the conveyor for the overall system (see 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.

C0 consists of test cases with no failure as comparative data. C1 consists of test cases with 5 conveyor failures in each test case, which needs between 1,757 – 12,672 milliseconds to resolve failures. C2 consists of test cases with 2 machine failures in each test case, which needs between 5,942 – 18,195 milliseconds to resolve failures. C3 consists of test cases with combination of 5 conveyor failures and 2 machine failures in each test case, which needs between 1,341 – 17,744 milliseconds to resolve failures. The machine failures and the conveyor failures occur randomly in the test cases.

Table 1. Failure Classes and Risk Analysis

Failure Class	Failure Impact
C0	No failure
C1	Conveyor failure
C2	Machine failure
C3	Combined conveyor failure and machine failure

In our research, we perform two different kind of process analysis to find out the relationships between information from two heterogeneous layers, namely business layer and process layer. We apply a process model approach to show the actual model of processes running in the system by analyzing the event log of the simulation. We also apply an organizational mining approach to show the relationships between machines used in the simulator. This information is useful to discover the usage frequency of machines and how machines are related, so we can suggest maintenance intervals for machines and also how machines can be replaced by other machines with similar functions.

5 Results

In this section, we describe the design and results of the process analysis and evaluation in the context of the production automation systems, with SAW simulator as a use case. We selected the SAW simulator because of the involvement of a set of different layers (business, process, and machine layers) as a representative of a real production automation system. The evaluation follows typical process analysis and mining guidelines [21] to organize, plan and execute the evaluation study. First, we collect activities information in the events and transformations needed to change the event into required format for process analysis tool, e.g., ProM. Second, we perform statistical analysis on the collected events to count the products finished for each failure class and frequency of machine used in the system. Third, we analyze the process model of the events. Fourth, we perform an organizational analysis for discovering the structure and relationships of machines used in the SAW simulator.

5.1 Event Log Transformation for Process Analysis

The SAW event log is the foundation for performing process analyses. The event log is generated by the SAW event engine in the process layer. The structure of event logs produced by the SAW event engine is illustrated in Figure 2.

```
<eventlog>
        <testrun id="1">
                <event id="21" timestamp="5700" type="evtWorkpieceOut">
                        <payload key="OrderId">1</payload>
                        <payload key="WorkpieceId">SW003</payload>
                        <payload key="ComponentName">DS1</payload>
                </event>
                …
        </testrun>
</eventlog>
```

Figure 2. Structure of SAW event logs

Activities of each component of the SAW simulator are recorded as event logs 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.

Process mining 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. 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. An example of transformation result can be seen in Figure 3.

```
<WorkflowLog>
<Source program="SAW"/>
<Process id="1">
<ProcessInstance id="1">
<AuditTrailEntry>
        <EventType unknowntype="evtWorkpieceOut">unknown</EventType>
        <WorkflowModelElement>SW003</WorkflowModelElement>
        <Originator>DS1</Originator>
        <Timestamp>2010-06-10T16:15:32.25+02:00</Timestamp>
</AuditTrailEntry>
…
```


Figure 3. Transformation of event logs for further process analyses

The MXML (Mining XML) format is the format used by ProM to mine and analyze the event data. The MXML format consists of two parts, the source and processes. Each process consists of process instances that contain detailed audit trail entries. Each audit trail entry consists of a workflow model element, an event type, a timestamp and an originator.

The transformation processes from the SAW simulator event file to the process analysis tool format (MXML) are done by using XSLT (Extensible Stylesheet Language Transformations) file. The transformation processes are as follows. (1)Adapt the time by adding the timestamp in millisecond to the base time (the real time at that moment) and change the format to time format (YYYY-MM-DDTHH:MM:SS). (2) For each work order, we use it as a process instance and we aggregate all events belonging to a single order. (3) Sorting event log with the same order id from test run data into a single process instance. (4) SAW simulator event types are adapted into event type format. (5)The component names of the simulator event are transformed into the originator of the event. (6) The work piece identifier of the simulator event is transformed into the workflow model element.

5.2 Statistical Analysis

To show the use case data, we perform statistical analysis on collected events as follows. Figure 4 illustrate 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. 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.

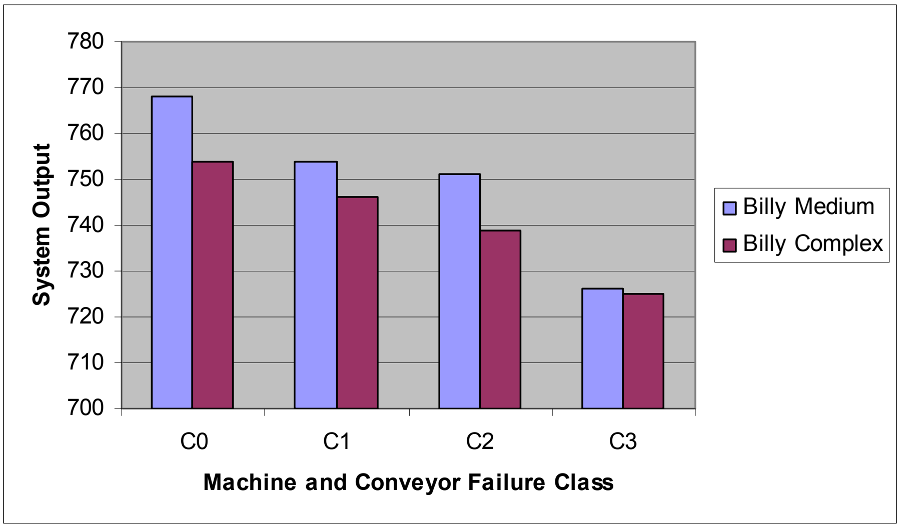

Figure 4. Relationships between failure classes and system output.

5.3 Process Model

Figure 5 shows the relationship between the process models which are formed based on the application of the process analysis tool to the event logs and the pre-defined product trees of certain products. A product tree consists of description of products, the parts to build the product and machine function that build the product from its parts. Here we use two product trees for two different product types, namely Billy Medium and Billy Complex. 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 produced by analyzing event logs using process analysis tool called ProM. The model can show 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 check conformance of the process model to the product trees.

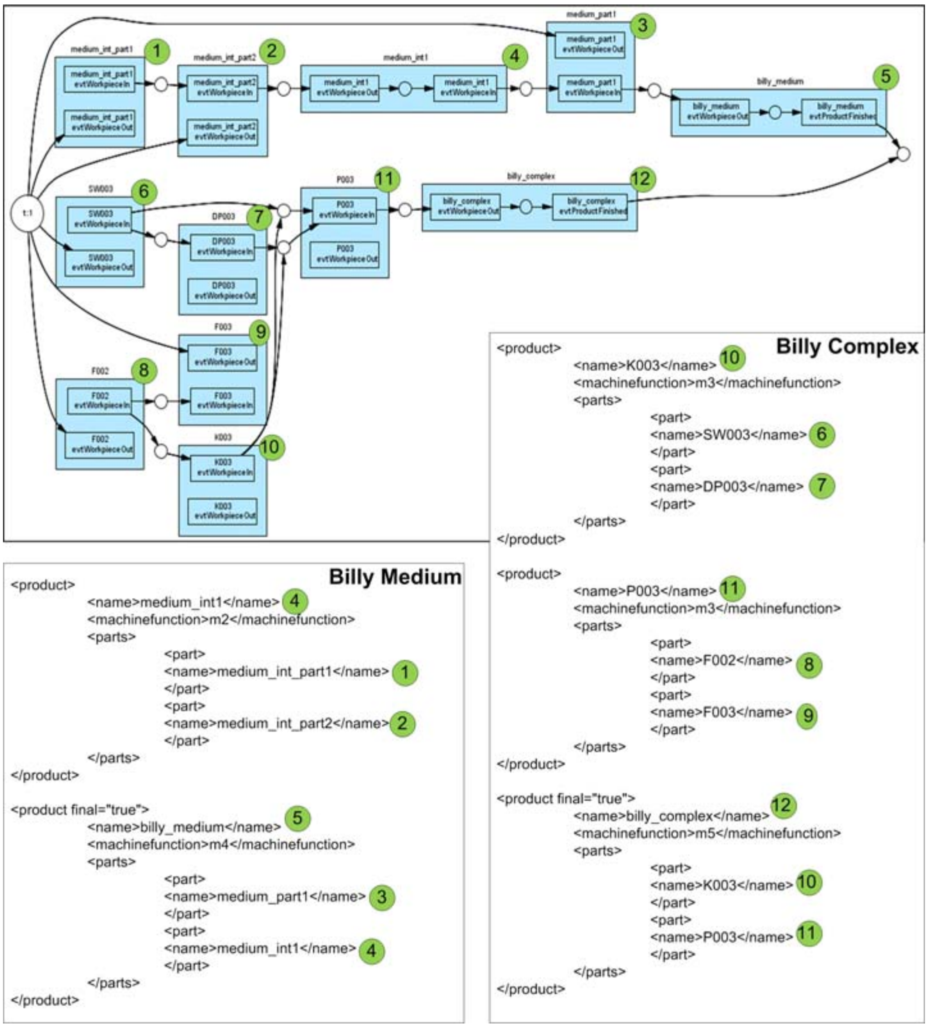

Figure 5. Relationship between process model and product trees

5.4 Organizational Mining

Organizational model mining aims at deriving organizational models from process logs. There are two kinds of models, namely (1) the organizational model that represents the current organizational structure and (2) the social network that shows the communication structure in an organization. An organizational model usually consists of organizational units (e.g., functional units), roles (e.g., duty), originators, and their relationships (e.g., who plays what roles, who belongs to which functional unit, what is the hierarchy among organizational units). A social network is a network in which nodes represent individuals or organizational units, and arcs between the nodes represent the relationships between them. The generated social networks allow organizations to monitor how people and groups work together. In the SAW simulator context, the organizational model is needed to show the actual organizational structure from the run time information, while the social network can be used to show how machines in the simulator are connected and work together.

In this paper, we make organizational models that show the relationships between machines that do similar tasks and the relationships between machines that work together to produce some products. Methods to show these relationships are inspired by the metrics that based on joint activities/doing similar tasks and the metrics that based on joint cases/working together as proposed in [19].

Metrics on doing similar tasks focus on the activities that individuals perform. We assume that originators doing similar things are more closely linked than originators doing completely different tasks. Each originator has a profile (i.e., originator by activity matrix) based on how frequently they conduct specific activities. From the profile, we can measure the distance between the profiles of different originators by comparing the corresponding row vectors. In this paper, we calculate Pearson's correlation coefficient to quantify this distance. The Pearson's correlation coefficient uses values ranging from -1 to +1. Since the positive values imply positive linear relationships between variables, we applied the threshold value of 0.0 and removed negative arcs from the network.

Figure 6 shows the graph of the "doing similar tasks" metric based on the SAW event logs. The graph illustrates the relationships between originators of events, i.e. the machines. The numbers on arcs specify the correlation coefficient between two machines as nodes. The higher this correlation coefficient is, the more similar two machines are. Based on this graph, we find that DS3 and DS4 are machines with similar functions. So are DS5 and DS6. They have the highest correlation coefficient (nearly 1) and in fact they are similar machines.

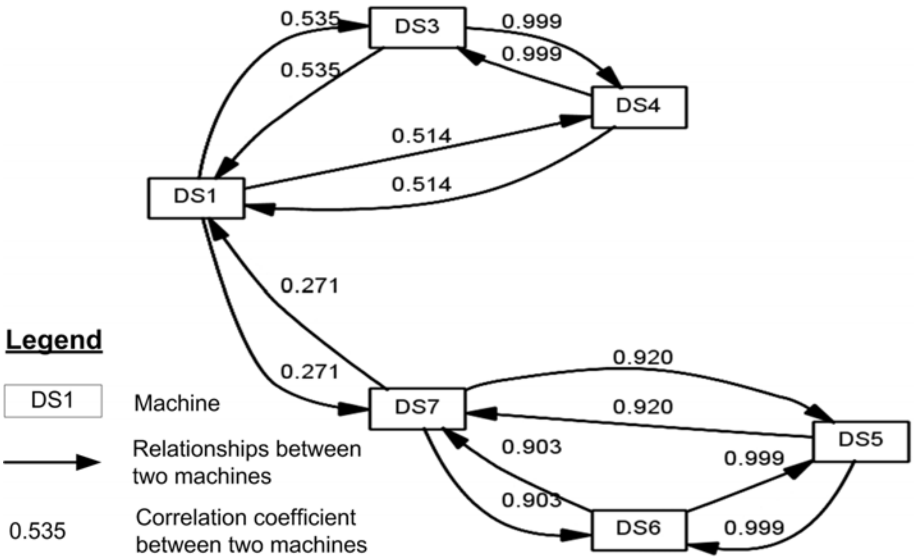

Figure 6. Organizational model of relationships between machines that do similar tasks.

The “working together” metric counts how frequent two originators are performing activities on the same product. If originators work together on product, they will have a stronger bond than originators who rarely work together.

Figure 7 shows the graph of the “working together” metric based on the SAW event logs. The graph illustrates the relationships between the machines. The numbers on the arcs specify the ratio of machines participation on producing a product. For example, suppose that machine DS6 participates in four products, machine DS1 participates in eight products, and they work together four times. In this situation, DS6 always work together with DS1, but DS1 does not. Thus, the value for DS6 to DS1 (which is 1.000) has to be larger than the value for DS1 to DS6 (which is 0.500).

From figure 7, we find that the machine DS1 has relationships with all other machines. DS5 and DS6 are similar machines that don’t work together, while even DS3 and DS4 are similar machines, they work together for some tasks.

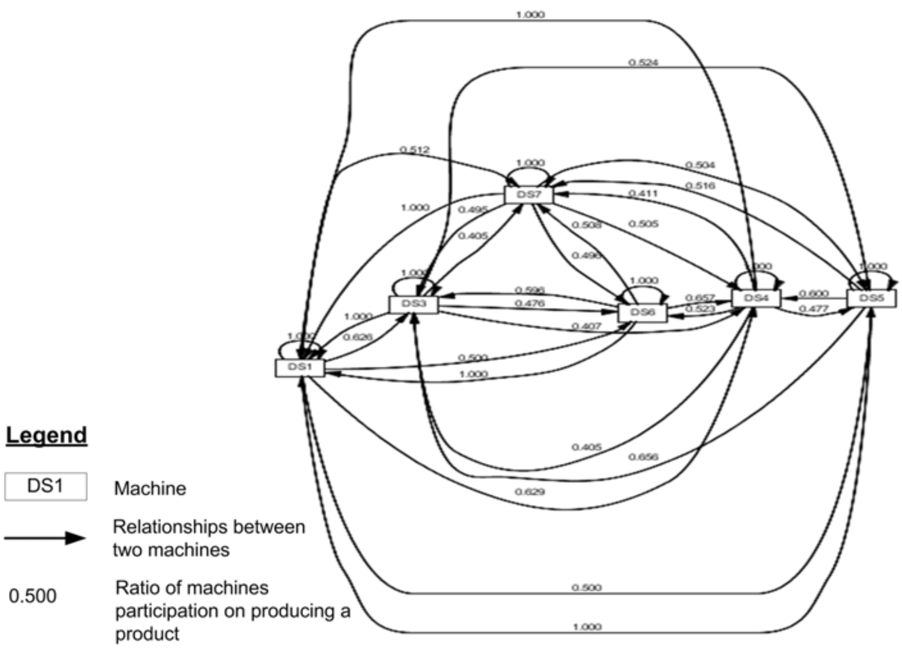

Figure 7. Organizational model of relationships between machines that work together on the same product.

6 Discussion and Conclusion

In this paper, we have addressed the using of run time information, i.e. process events, in production automation systems for improving systems productivity and effectiveness. We used the SAW simulator as a use case for the production automation system such that the analysis approach applied in this use case is possible to be generalized in other systems.

A previous research on improving production automation systems has been done by doing performance evaluation on 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. However, this approach was not considering the maintenance efforts of the system components.

In this paper, we used process analysis and organizational mining to explore the information from event log collected during the simulator run time. By using process analysis we can derive process models that illustrate the flow of process 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.

By using an organizational mining approach, we can discover the organizational structure of the machines using the information from the collected event logs. We found that the organizational mining can help us to know the structure of machines in the layout and their relationships during run time.

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. By using process analysis, we can also correlate the product trees and the actual process models and inspect whether any deviation happens between two models.

The organizational mining can show the relationships of machines with similar function and the relationships of machines that work together. By using organizational mining on process event logs, we can discover a new relationship of originator at run time that is not easily found in normal way, e.g., whether a machine is working together with other machine or not, even though both machines are similar.

However, the current evaluation is more focused on machines structures and relationships, while interactions with other components of the systems, e.g., conveyors and junctions are not discussed. We can extend the organizational mining approach for other components in other engineering domains as well.

Future work will be to analyze challenging engineering processes and environments by using a Bayesian Network approach 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.

Acknowledgments. 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.

References

1. Baker, P., Zhen, R.D., Gabrowksi, J., Oystein, H.: Model-Driven Testing: Using the UML Testing Profile. Springer (2008)
2. Biffl, S., Sunindyo, W., Moser, T.: Semantic Integration of Heterogeneous Data Sources for Monitoring Frequent-Release Software Projects. In: 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2010), pp. 360-367, IEEE Computer Society, Krakow, Poland, (2010)
3. Fenton, N., Neil, M.: A Critique of Software Defect Prediction Models. IEEE Transactions on Software Engineering 25, 675-689 (1999)
4. GAMP 5: Good Automated Manufacturing Practice. International Society for Pharmaceutical Engineering (ISPE) (2008)
5. Gerke, K., Cardoso, J., Claus, A.: Measuring the Compliance of Processes with Reference Models. On the Move to Meaningful Internet Systems: OTM 2009 (2009) 76-93
6. Gruber, T.: Towards principles for the design of ontologies used for knowledge sharing. Int. J. Human-Computer Studies 43, (1995)
7. Lastra, J.L.M., Delamer, I.M., Ubis, F.: Domain Ontologies for Reasoning Machines in Factory Automation. ISA o3neida (2009)
8. Li, P.L., Herbsleb, J., Shaw, M.: Forecasting Field Defect Rates Using a Combined Time-Based and Metrics-Based Approach: A Case Study of OpenBSD. In: 16th IEEE International Symposium on Software Reliability Engineering, pp. IEEE Computer Society, (2005)
9. Lüder, A., Peschke, J., Reinelt, D.: Possibilities and Limitations of the Application of Agent Systems in Control. In: International Conference On Concurrent Enterprising (ICE), (2006)
10. Merdan, M., Moser, T., Wahyudin, D., Biffl, S.: Performance evaluation of workflow scheduling strategies considering transportation times and conveyor failures. In: IEEE International Conference on Industrial Engineering and Engineering Management 2008 (IEEM 2008) pp. 389-394, (2008)
11. Merdan, M., Moser, T., Wahyudin, D., Biffl, S.: Performance evaluation of workflow scheduling strategies considering transportation times and conveyor failures. In: IEEE International Conference on Industrial Engineering and Engineering Management (IEEM 2008), pp. 389-394, (2008)
12. Mockus, A., Weiss, D., Zhang, P.: Understanding and Predicting Effort in Software Projects. In: International Conference on Software Engineering (ICSE), pp. (2003)
13. Noy, N.F.: Semantic integration: a survey of ontology-based approaches. SIGMOD Rec. 33, 65-70 (2004)
14. Noy, N.F., Doan, A.H., Halevy, A.Y.: Semantic Integration. AI Magazine 26, 7-10 (2005)
15. Obitko, M., Marik, V.: Ontologies for Multi-Agent Systems in Manufacturing Domain. In: 13th International Workshop on Database and Expert Systems Applications, pp. IEEE Computer Society, (2002)

16. Rembert, A.J., Ellis, C.: An initial approach to mining multiple perspectives of a business process. In: 5th Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations, pp. 35-40, ACM, Portland, Oregon, (2009)
17. Schäfer, W., Wehrheim, H.: The Challenges of Building Advanced Mechatronic Systems. In: 2007 Future of Software Engineering - International Conference on Software Engineering, pp. 72-84, IEEE Computer Society, Washington, DC, (2007)
18. Song, M., van der Aalst, W.M.P.: Towards comprehensive support for organizational mining. Decis. Support Syst. 46, 300-317 (2008)
19. van der Aalst, W.M.P., Reijers, H.A., Song, M.: Discovering Social Networks from Event Logs. Computer Supported Cooperative Work 14, 549-593 (2005)
20. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster., L.: Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering 16, 1128-1142 (2004)
21. van Dongen, B.F., van der Aalst, W.M.P.: A Meta Model for Process Mining Data. In: CAiSE'05 WORKSHOPS, pp. 309-320, (2005)
22. Vrba, P.: MAST: Manufacturing Agent Simulation Tool. In: Emerging Technologies and Factory Automation, pp. 282-287, IEEE Computer Society, (2003)
23. Wahyudin, D., Schatten, A., Winkler, D., Tjoa, A., Biffl, S.: Defect Prediction using Combined Product and Project Metrics: A Case Study from the Open Source "Apache" MyFaces Project Family. In: Proceedings of the Software Engineering and Advanced Applications, 2008. SEAA '08. 34th Euromicro Conference, pp. 207 - 215, IEEE Computer Society, (2008a)
24. Winkler, D., Biffl, S., Östreicher, T.: Test-Driven Automation – Adopting Test-First Development to Improve Automation Systems Engineering Processes. In: 16th EuroSPI Conference, pp. to appear, Madrid, Spain, (2009)