TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

NAVID KHAJEHZADEH

DATA AND PROCESS MINING APPLICATIONS ON A MULTI-CELL FACTORY AUTOMATION TESTBED

MASTER OF SCIENCE THESIS

Examiner: Professor Jose L. Martinez Lastra

Examiner and topic approved in the Automation, Mechanical and Materials Engineering Faculty Council Meeting on 7 November 2012.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master of Science Degree Programme in Machine Automation
KHAJEHZADEH, NAVID: Data and Process Mining Applications on a Multi-Cell Factory Automation Testbed
Master of Science Thesis, 66 pages, 3 Appendix pages
December 2012
Major subject: Factory Automation
Examiner: Professor Jose L. Martinez Lastra
Keywords: energy awareness, behavioural patterns, monitoring, energy consumption, factory automation, fault detection, process mining, model discovery, event logs, information system

This paper presents applications of both data mining and process mining in a factory automation testbed. It mainly concentrates on the Manufacturing Execution System (MES) level of production hierarchy.

Unexpected failures might lead to vast losses on investment or irrecoverable damages. Predictive maintenance techniques, active/passive, have shown high potential of preventing such detriments. Condition monitoring of target pieces of equipment beside defined thresholds forms basis of the prediction. However, monitored parameters must be independent of environment changes, e.g. vibration of transportation equipments such as conveyor systems is variable to workload. This work aims to propose and demonstrate an approach to identify incipient faults of the transportation systems in discrete manufacturing settings. The method correlates energy consumption of the described devices with the workloads. At runtime, machine learning is used to classify the input energy data into two pattern descriptions. Consecutive mismatches between the output of the classifier and the workloads observed in real time indicate possibility of incipient failure at device level.

Currently, as a result of high interaction between information systems and operational processes, and due to increase in the number of embedded heterogeneous resources, information systems generate unstructured and massive amount of events. Organizations have shown difficulties to deal with such an unstructured and huge amount of data. Process mining as a new research area has shown strong capabilities to overcome such problems. It applies both process modelling and data mining techniques to extract knowledge from data by discovering models from the event logs. Although process mining is recognised mostly as a business-oriented technique and recognised as a complementary of Business Process Management (BPM) systems, in this paper, capabilities of process mining are exploited on a factory automation testbed. Multiple perspectives of process mining is employed on the event logs produced by deploying Service Oriented Architecture through Web Services in a real multi-robot factory automation industrial testbed, originally used for assembly of mobile phones.

PREFACE

This thesis work was accomplished at the Factory Automation Systems and Technologies Lab (FAST), Department of Production Engineering at Tampere University of Technology, under direction of Professor Jose L. Martinez Lastra and Research fellow Corina Postelnicu.

Funding of this work came from AREMIS Second Call 2009 Programme under Agreement Number 100223, correspondent to the project shortly entitled eSONIA: Embedded Service Oriented Monitoring, Diagnostics and control: Towards the Asset Aware and Self Recovery Factory.

I would like to express my sincere appreciation to Prof. Jose L. Martinez Lastra for giving me the opportunity to study at TUT and to work in FAST laboratory which I always remember it as a nice, well-equipped and friendly multinational place.

I would like to extend my appreciation to Dr. Andrei Lobov for preparing and giving such a high-level education as well as lots of joys and happiness.

My heartfelt gratitude goes to my supervisor, Dr. Corina Postelnicu for her infinite support, help and professional guidance, reviews and comments. Without her constructive supervision it would be very difficult for me to finish this thesis.

I would like to thank all my friends and colleagues in TUT and FAST laboratory especially Tomas, Luis, Hector and Bin for encouraging me all the times when I needed motivation and for making an enjoyable place to study and work together.

A world of thanks to my parents for their endless support and unconditional love and help. Thanks to my brothers and sister for always persuading me to aim for the highest in my life and to never surrender to the problems.

ACRONYMS

IS	Information System
BPM	Business Process Management
CRM	Customer Relationship Management
ICT	Information and Communication Technology
PAIS	Process-Aware Information Systems
WFM	Work-Flow Management
BPMN	Business Process Model and Notation
EPC	Event Driven Process Chains
UML	Unified Modelling Language
PN	Petri Net
WFN	Work Flow Net
YAWL	Yet Another Workflow Language
IT	Information Technology
C-net	Causal net
FSM	Finite state machine
GA	Genetic Algorithm
HM	Heuristic Miner
BI	business intelligence
BAM	Business Activity Monitoring
CEP	Complex Event Processing
CPM	Corporate Performance Management
CPI	Continuous Process Improvement
BPI	Business Process Improvement
TQM	Total Quality Management
KPI	Key Performance Indicators
SEMMA	Sample, Explore, Modify, Model and Assess
LQN	Layered Queuing Networks
PCA	Principal Component Analysis
PLS	Partial Least Squares
SVM	Support Vector Machines
LS-SVM	Least Squares Support Vector Machines
QP	Quadratic Programming
SAW	Simulator of Assembly Workshops
LTL	Linear Temporal Logic
MXML	Magic eXtensible Markup Language
WS	Web Services
DPWS	Device Profile for Web Services
RTU	Remote Terminal Unit
MES	Manufacturing Execution Systems
DSS	Decision Support System

SOA	Service Oriented Architecture
CRISP-DM	Cross Industry Standard Process for Data Mining

LIST OF FIGURES

Figure 1. Application possibilities of process mining in factory automation 13
Figure 2. A general structure for producing event logs 15
Figure 3. A sample discovered model for a Lasagna process 18
Figure 4. A sample discovered model for a Spaghetti process 18
Figure 5. Process mining overview [11] 21
Figure 6. Concrete activities in process mining life cycle [53]..22
Figure 7. BPM life-cycle 24
Figure 8. Classification of process history-based methods..25
Figure 9. Mapping non-linearly separable data into a higher dimensional space 27
Figure 10. Amodel representing steps to produce a work-flow model by alpha-algorithm.......................28
Figure 11. Production automaton system structure 33
Figure 12. FASTory Line 38
Figure 13. A detailed view of one cell of FASTory line 38
Figure 14. Layout of FASTory line from top view 39
Figure 15. Conveyor system analyzed. Main conveyor hosting 2 pallets. Bypass hosting 1 pallet............40
Figure 16. The previous system architecture for FASTORY line...41
Figure 17. Architecture of retrofitted FASTORY line 42
Figure 18. Setting used to monitor energy consumption...44
Figure 19. Cell 5 bypass conveyor engine power consumption..46
Figure 20. Energy awareness for detection of gradual conveyor missalignment......................................47
Figure 21. Cell 5 conveyor system engine (main and bypass conveyor) power consumption...................47
Figure 22. Classes generated by rule based engine and correlated to each sampled data..........................48
Figure 23. Classified cross validation data generated by LS_SVM 48
Figure 24. ISA-95 architecture focusing on level 3 50
Figure 25. Applied method for storing the data including necessary parameters 50
Figure 26. Fragment of generated event log by XESame tool...53
Figure 27. Alpha algorithm applied to Fastory event log..54
Figure 28. Feasible message sequence patterns between consecutive cells in Fastory..............................55
Figure 29. ProM's Heuristic Miner applied to the event log of Fastory (Partial result). 56
Figure 30. The Heuristic model of Figure 24, converted to Petri Nets...56
Figure 31. Pie chart, basic performance analysis...57
Figure 32. Basic performance analysis (x axis - performers; y axis – total cell working time, seconds) ..57
Figure 33. KPIs including IPC-2541 states overview 58
Figure 34. Pallets performance. Dotted chart analysis 60
Figure 35. The effect of improper operation of pallet 33 on other pallets...60
Figure 36. a : The Conformance Checker b: zoom view with descriptions..62
Figure 37 . log diagnostic perspective..62
Figure 38. Noncompliance between the model and the log (highlighted in orange).................................63
Figure 39. log view, visualizes each process instance...64
Figure 40.Unusual behaviour observed in the generated model from the log...65
Figure 41. Incorrect transition between cell 3 and cell 4 of trace number 100 66
Figure 42. Dotted Chart implementation for cellNumber 3 messages 66
Figure 43. MXML log format...74
Figure 44 . XES log format 75

LIST OF TABLES

Table 1. Popular Process Modelling Paradigms 17
Table 2. Process mining tools, short view...35
Table 3. ProM plugins applicable to process discover 36
Table 4. ProM plugins applicable to check conformance...37
Table 5. Testbed generated messages 42
Table 6. Possible control scenarios considered for FASTory line 43
Table 7. Defined rules of event processing 51
Table 8. Raw data associated with producing one product stored in MySQL database 52
Table 9. Applied XES Standard extensions...52
Table 10. Comparison of online KPIs with KPIs applied offline for basic performance analysis 58
Table 11. Pallet performance (Basic Pefrormance Analysis plugin, text view)..59

TABLE OF CONTENTS

Abstract 1
PREFACE 2
ACRONYMS 3
LIST OF FIGURES 5
LIST OF TABLES 6
TABLE OF CONTENTS 7
1. INTRODUCTION 9
1.1. Problem Definition 9
1.1.1. Problem statement 9
1.1.2. Justification of the work 10
1.1.2.1 Fault detection and diagnosis 10
1.1.2.2 Process Mining for Discovery 10
1.2. Work description 11
1.2.1. Objectives 11
1.2.2. Methodology 11
1.3. Thesis outline 12
2. BACKGROUND 13
2.1. Information Systems 14
2.2. Event Logs 15
2.3. Process Models 16
2.3.1. General applications of models 16
2.3.2. Lasagna versus Spaghetti processes 17
2.4. Data and process mining 18
2.4.1. Data mining 18
2.4.2. Process mining 20
2.4.3. Data vs. process mining 24
2.5. Data Mining Application: Fault detection and diagnosis 25
2.5.1. Support Vector Machines (SVM) 26
2.5.2. Least Squares Support Vector Machines 27
2.6. Process Mining Applications 27
2.6.1. Process Mining for Discovery 27
2.6.1.1 Discovery based on alpha algorithm 28
2.6.1.2 Discovery based on the heuristic approach 29
2.6.2. Basic Performance Analysis 30
2.6.3. Process mining in practice: four related studies 32
2.6.4. Available toolkits 34
2.6.4.1 ProM Framework Tool and relevant Plugins 35
3. TESTBED 38

4. DATA AND PROCESS MINING IN PRACTICE: RESULTS 45
4.1. Data mining for fault detection and diagnosis: Incipient deterioration in transportation equipment segments 45
4.1.1. Experiment of correlation between power consumption values and workload 45
4.1.2. Proposed approach for fault detection 46
4.1.3. Empirical results of the proposed approach 47
4.2. Application of process mining in factory automation 49
4.2.1. Data collection and event log preparation (inspection/cleaning) .. 50
4.2.2. Process Mining for Discovery: Results 53
4.2.2.1 Discovery based on the alpha algorithm 54
4.2.2.2 Discovery based on the heuristic approach 55
4.2.3. Basic Performance Analysis 57
4.2.3.1 General performance of the line 57
4.2.3.2 Performance analysis of the pallets 59
4.2.4. Process Mining for Conformance: Results 61
4.2.4.1 Enhancement of the discovered model 61
4.2.4.2 Process diagnosis and enhancement 64
References 68
APPENDIX: MXML versus XES format of event logs 73

1. INTRODUCTION

1.1. Problem Definition

1.1.1. Problem statement

In discrete manufacturing settings, adjacent conveyor pieces tend to exhibit misalignment in time. The extent to which device segments drift away from each other depends on many factors, including the nature of the production process, pallet weight, total transfer time for a pallet, etc. Predictive maintenance techniques revealed to be extremely helpful to estimate the deterioration of such problems. Data traditionally used in predictive maintenance (vibration, temperature, pressure and humidity) are inclined to be influenced by environmental changes. For instance, vibrations of conveyor systems are prone to be affected by workload of conveyor.

Data-driven fault detection is based on historical observations of process data. Abnormal system behaviour is signalled via mathematical or statistical algorithms, e.g. neural networks or machine learning techniques e.g. Support Vector Machines. The main advantage of the data-driven approach over model-based fault detection is that an accurate mathematical modelling of the system is not essential. The main challenge here is the need for large quantities of training data of good quality.

Affiliation of condition monitoring parameters with surrounding changes might be seen as a disadvantage; however it may be feasible to benefit from such dependences. Association of the energy consumption information with particular workload may be a promising way to detect gradual undesired behavioural changes in the considered equipment piece.

Data mining techniques either cover only a few sections of the processes or elicit abstract patterns in terms of rules or decision trees. It is not possible to infer global optimal functioning of factory floor processes by summing up inspection results of many local sections. Process mining combines data mining and process modelling to generate models from IT log data. Process mining considers sets of data events as process instances. Having process instances, it is feasible to evaluate processes from a wider scope, e.g., organizational, case and process perspectives.

Despite its being a young research area, process mining has demonstrated capabilities to overcome the difficulties arising with large amounts of unstructured data and events produced by heterogeneous resources associated with information systems at all levels.

Some facts about the systems of interest are not immediately obvious. Hidden patterns are discoverable in the immense amount of data such systems are breathing in and out. Inferences are done based on historical process information. This might include time, energy, workload, social network, dependency of activities, resources, states of components, embedded rules, flow of activities, etc.

Based on the discovered information, process models for such systems can be improved in real time. Comparison of real data against initial process models may lead to conclusions regarding possible faults and failures in the systems analysed, and in some cases leads to enactment of further processes.

1.1.2. Justification of the work

1.1.2.1 Fault detection and diagnosis

Small failures can lead to significant financial losses or hazardous situations. Faults have been e.g. responsible for 3% to 8% decrease in oil production, causing up to $20 billion losses in the US economy [82]. Early detection of faults is therefore critical to prevent serious disruptions to the production process. In addition to vibration, temperature, pressure and humidity data traditionally used in predictive maintenance [83], energy consumption signatures of pieces of equipment are a promising way to detect faults that occur gradually. An example of such faults is the misalignments of conveyor segments that generally occur as time passes in discrete manufacturing execution systems due to e.g. friction.

In the context of predictive maintenance, failure thresholds are defined by experiencing repeatedly equipment failures. This is hazardous and expensive. Common parameters widely applied for maintenance are prone to be influenced by environmental changes especially for transportation equipment. For instance, the workload of a conveyor system is directly affecting the parameter vibration. Therefore a smart method which associates environment alterations to measured parameter is needed. In this work we present an approach which links the workload of conveyor to its engine power consumption and employ it for early fault detection.

1.1.2.2 Process Mining for Discovery

Considering the high rate of advances in industrial technologies, companies need to adjust their processes at a faster pace. Process mining provides suitable algorithms and techniques to overcome the difficulties of process analysis and support any necessary modifications.

When there is no official model explaining the behaviour of processes or when there is not a complete and revealing documentation, process mining techniques are able to provide the requirements.

In multi-agent production lines consisting of numerous pallets, tracking all pallets individually in order to discover malfunctions (e.g. bearing damages, lubrication) is expensive and time consuming, and prone to errors. Process modelling together with mining techniques may provide valuable insights about the performance of processes.

1.2. Work description

1.2.1. Objectives

The objectives of this work are:

1. Proposing an approach in the context of data mining as a component of process mining, using machine learning (data classification), pointing to failure of transportation devices based on energy consumption patterns [3],[4].
2. To produce event logs relevant to a representative multi-robot factory automation testbed.
3. To convert the event logs to a format compatible with process mining tools and to generate parameters required by process analyzer
4. To apply process mining techniques on the produced logs in order to discover hidden patterns of behaviour (e.g. analyse the performance of production line segments, discover a model reflecting the real behaviour of the processes, analyzing the workload balance of the line, adding/removing missed/extra events, analyze the performance of pallets)
5. To compare the produced logs and models to the reference process models (rules), in order to:
    a. Detect problems in the designed process model or/and discovering those parts not feasible based on real behaviour of the processes
    b. Detect problems in the communication of factory automation systems
    c. Improve the discovered model in order to provide a complete report reflecting process behaviours.

1.2.2. Methodology

In data mining and fault detection part of this work, power consumption values coming in real time from the line are labelled by the concerned workload on the conveyor system at the same time. Behaviour of tesbed pieces of equipment from energy consumption point of view is characterised using a supervised machine learning classification algorithm (SVM). A rule based engine is defined offline and integrated in the proposed model and finally an approach able to detect malfunctioning of transportation pieces of equipments based on their energy consumption signatures is proposed.

Given the process mining part of this thesis work, event logs are produced from the data coming from the controllers embedded in a production line; they are converted to required XML formats. The state of the art of multiple tools having capabilities of process mining is reviewed. Worthwhile insights extracted from the event logs and performance of the production line is evaluated. Having assessed multiple discovery algorithms, process modelling is performed using heuristic approach and the models are represented by C-net models. Using some conformance checking techniques, the real behaviour of the system is compared with the expected behaviour, finally a number of faults existed in the communication part of the line is discovered.

1.3. Thesis outline

Chapter 2 introduces and explains the background knowledge about data mining and process mining, theories, state of the art, applications and tools utilizable in factory automation. Chapter 3 explains the testbed and integrated components, employed tools and the way energy monitoring is performed. Chapter 4 documents the proposed approach for detection of conveyor misalignment and demonstrates usability of process mining techniques in a factory automation testbed and conclusions in chapter 5.

2. BACKGROUND

This chapter starts by explaining steps required in order to employ process mining techniques in factory automation.

An information system provides event logs over data captured from the process layer. In the context of process mining, Complex Event Processing (CEP) is a typical technique applied by IS in order to provide necessary parameters. Event logs are stored in a database, in various formats (MXML, XES, etc.) compatible with the desired target process mining tools (e.g. ProM Framework [1]). Process mining techniques for Work-flow discovery are applied on the recorded data to output a model of the transition of activities between resources. Inspection of this resulted model and conformance checking of a reference model against the event log leads to e.g. improvements to the reference model, failure detection and diagnostics in pieces of equipment, or conclusions about the performance of the system. Figure 1 illustrates the application possibilities of process mining.

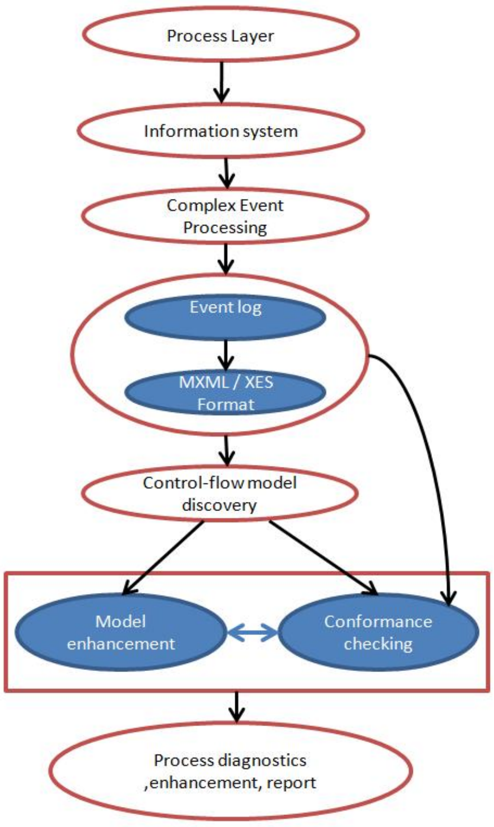


***Figure 1**. Application possibilities of process mining in factory automation*

2.1. Information Systems

An information system refers to the interconnection and interaction of IT (information technologies, data and processes) and people's activities in order to support decision making at management and operation levels [5], [6].

Information systems log data and events to expose the internal functioning of a system to the outside world. General functions of an IS include the tracking, supervision and control of (business) processes. Information systems are of many types, depending on the target domain and purpose:

Work Flow Management (WFM) is a system which supervises the process of transferring the process information, activities, tasks or responsibilities from one machine or employee to another one. Appropriate implementation of workflow system ensures the shift of work compatible with some predefined procedures. WFM performs both automating redundant tasks and responsibilities and also controlling the automated processes [98].

Business Process Management (BPM) [7], [8] combines IT and management regulations to analyze and manage, as well as automate process activities. BPM is an extension of Work Flow Management (WFM).

Process-Aware Information Systems (PAISs) go beyond traditional WFM by providing and supporting more explicit tasks, consciously engaging in the management and/or coordination of processes of interest. ERP systems like SAP and Oracle, Customer Relationship Management (CRM) systems, rule-based systems, call centre software, high-end middleware (WebSphere) are some examples of information systems that are aware of the processes that they support.

Today's ISs are handling huge amounts of diverse data. Classical Work-Flow Management (WFM) systems (e.g., Staffware[9] and COSA[10]), BPM systems (e.g., BPM by Pallas Athena, SmartBPM by Pegasystems, FileNet, Global 360, and Teamwork by Lombardi Software), ERP systems (e.g., SAP Business Suite, Oracle E-Business Suite, and Microsoft Dynamics NAV), PDM systems (e.g., Windchill), CRM systems (e.g., Microsoft Dynamics CRM and SalesForce), middleware (e.g., IBM's WebSphere and Cordys Business Operations Platform), and hospital information systems (e.g., Chipsoft and Siemens Soarian) are examples of information systems providing event logs which represents detailed information about the activities[11].

Enterprise IS relies on a structured format for data storage. Typically this format is an event log [16]. For instance, Work Flow Management Systems (WFMSs) typically record the start and completion of the activities[12], ERP systems like SAP register all transactions such as all the forms filled out by clients. B2B systems record the messages reciprocated between different sides of business. CRM systems log the messages associated with communications with customers[13],[14],[15].

2.2. Event Logs

Information systems from vendors like SAP or Oracle are able to handle everything concerning finance, resources, customer relationship management, track & trace within a company or factory. Whenever improvements / redesign of processes are needed, the starting point is to evaluate the execution of the processes. Most personnel responsible for inspecting /analyzing process executions via recorded data are software users rather than programmers. Raw data recorded in data bases often is not ready to be input to software applications.

There are several techniques for converting raw data or engineering information to a format clearly reflecting system executions. Such a format (the *event log*) must include all events related to every significant case, and information regarding execution date, time, originators (i.e. producers of tasks; in the context of this thesis originators are pallet IDs) and activities.

Event log generation is a highly complex task, requiring very good knowledge about the processes of interest and the significant events. Producing event logs is extremely dependent to decisions about how and what to extract.

The general mechanism of producing event logs is illustrated in *Figure 5*. The Process Definition stage is concerned with defining the structure of tasks or activities. An instance of a process is produced whenever a case (a set of activities performed in conjunction to fulfil some goals, e.g. activities performed by robots, conveyors and pallets to complete a product) is triggered, to keep track of cases and to provide traces of events within each case. Each event is an instance of an activity defined by process definition at the first layer. Events include a variety of attributes in order to provide enough information about the trace (e.g. date and time of execution, resources and executors of tasks, etc.).

Despite the general structure of the event log generation mechanism, resulting event logs depend on the different systems. Various decisions result in different logs and may lead to different analysis results. Projects must be analyzed with respect to the main process (es), and the rest should be neglected.

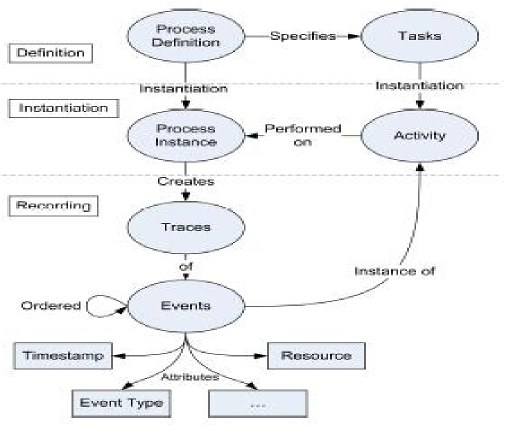


Figure 2. *A general structure for producing event logs [18]*

2.3. Process Models

Process mining aims at discovering, controlling and improving processes by generating models explaining the behaviour of systems and analyzing them. Process modelling is central part of process mining. This section discusses process models and classification of models based on their applications.

2.3.1. General applications of models

Process mining aims at producing a model extracted from the event logs by analyzing the events in order to explain the behaviour of the system. Purposes of generated process models include[11] :

- **Insight**: models are helpful for discussing about requirements, design decisions and confirming assumptions. Models lead the modeller to survey the model from different aspects, provide valuable insights and make components clear.
- **Discussion**: unquestionably used as the basis of discussions
- **Documentation**: Models are extremely applicable from educational point of view
- **Verification**: Applying models, possible errors inside the system can be perceived.
- **Performance analysis:** Different techniques are used to obtain a deep insight into the process. For instance, simulation reveals the causes of bottlenecks.
- **Animation**: Each process is controlled by a scenario. Animation is extremely useful for designers to get feedback from their control scenario design.
- **Specification:** models are extremely advantageous to explain PAISs in advance.
- **Configuration:** models can be used to configure a system.

Models can be classified according to several criteria:

First, depending on *how formal the model is*. With an informal model it is not possible to make a certain decision about the feasibility of a trace of activities; yet such models are utilized for structuring decisions or filing. A formal model, in contrast, provides sufficient support to decide about the possibility of a set of activities performed in sequence; at higher levels of BPM, organizations are keen to such models for analysis purposes and to enact operational processes via *play-out engines* (which permit only those activities which are allowed based on the model[99]).Semi-formal models are initially designed as informal, but during the process of implementation and interpretation subsets of them are formalized and supported by some formal and standard semantics. Examples include: Business Process Model and Notation (BPMN)[19] - a graphical representation applied for characterizing processes in a model; UML activity diagrams [20] - a standardized general-purpose modelling language including a set of graphic notation techniques to create visual models of object-oriented software-intensive systems; and Event Driven Process Chains(EPCs) [21] - flowcharts for business process modelling that are generally used for configuring an enterprise resource planning (ERP) implementation and for business process improvement.

Second, depending on *how the model is constructed* [22].Models may be created at design stage or may be derived from the system by reverse engineering or from event logs.

Table 1 lists modelling paradigms associated to processes for mining purposes, i.e. to retrieve 'hidden' information about the behaviour emerging at runtime.

***Table 1**. Popular Process Modelling Paradigms*

Transition systems	System models broken down to states and transitions. A fundamental process modelling paradigm. [11, page 31].
Petri nets	Places and transitions. Allows for dynamic analysis techniques grounded on a strong mathematical backbone. [11, page 33].
Work flow nets (WF-nets)	A subclass of Petri net. Modelling of life cycles of process instances. BPM and process mining [30][31][11].
YAWL (Yet Another Workflow Language)	Both a work flow modelling language and a BPM/Workflow system. YAWL workflow system handles complex data transformations, and full integration with organizational resources and external Web Services [32]. YAWL modelling language provides workflow patterns. A large number of patterns. The language is easy to use [11].
Business Process Modelling Notations (BPMN)	A standard process modelling paradigm. Graphical notations. Flowcharting. Similar to Unified Modelling Language (UML). BPMN can also demonstrate complex process semantics [11].
Event Driven Process Chains (EPCs)	Flow chart format for business process models. EPC provides business process models with a classical notation. The notation can be applied for configuring ERP implementations like ARIS and SAP R/3 [33], [11].
Casual nets (C-nets)	A causal net is a graph in which nodes illustrate activities and arcs represent causal dependencies. Causal nets are tailor-made for process mining. Results representation of several process discovery techniques (e.g., heuristic mining, fuzzy mining, and genetic mining). [11].

2.3.2. Lasagna versus Spaghetti processes

From the process mining point of view, processes are categorized according to their (un)structured nature:

In a *structured process*, the inputs and outputs are specified and clear, and all activities can be automated. Lasagna processes (*Figure 3*) are structured. Van Der Aalst [11] gives an informal definition for Lasagna process: "a process is a Lasagna process if with limited efforts it is possible to create an agreed-upon process model that has a fitness of at least 0.8" [11]. Almost all process mining algorithms can be applied on a Lasagna process.

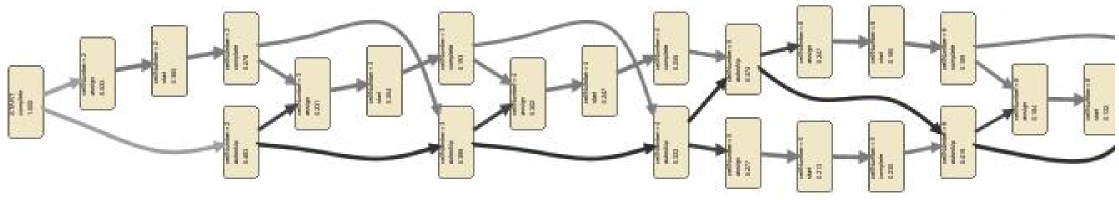


Figure 3. *A sample discovered model for a Lasagna process*

Semi-structured processes have an almost ordered structure, but they need still modification in order to have a transparent view. There is much room for deviation inside this category.

An *unstructured process* has an uncoordinated structure. Specifying an order for activities is challenging. They are not clearly expressed. They have a lot of small parts / details that are arranged in a complicated way and are therefore sometimes difficult to understand, solve or produce. Spaghetti processes (*Figure 4*) are unstructured. Few process mining techniques are applicable to them, sometimes the prerequisite being conversion of the unstructured process to Lasagna-like process models.

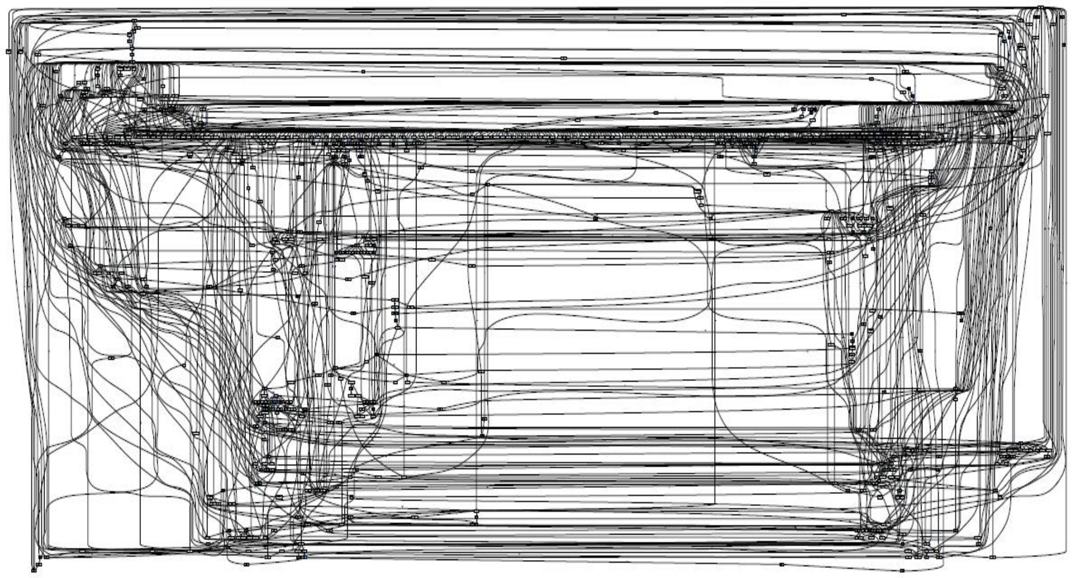


Figure 4. *A sample discovered model for a Spaghetti process [11]*

2.4. Data and process mining

2.4.1. Data mining

Data mining is defined as the procedure of extracting applicable, useful, inclusive information from databases[23] , [24] , [25] , [26]. Vapnik in [27] categorises data mining tasks as follows:

1) Class description: describing classes briefly but accurately

2) Association analysis: describing attribute or values that happen in a dataset frequently while there is correlation among them.

3) Cluster analysis: dividing data as objects into groups while objects of each group have similar characteristics. Clustering considered as the main task for explorative data mining. In clustering data are not labelled at training time and they are clustered based on optimizing interclass similarity rules.

4) Outlier analysis: Outliers are objects which behave far from the mainstream of the data.

5) Evolution analysis: Explains and models objects whose behaviours drifts over time. Time series, pattern matching and similarity are the main analytical techniques in this field.

Several different life-cycle and operation steps have been proposed for data mining:

The Cross Industry Standard Process for Data Mining (CRISP-DM) [50],[54] , defines a 6-step methodology for mining data and processes (understanding the business and the data associated to it, preparing the data, modelling, evaluation and deployment).

The Sample, Explore, Modify, Model and Assess (SEMMA) methodology[51]was proposed by the SAS institute to provide a list of activities for implementation of data mining. SEMMA has been defined to focus on the modelling part of the data mining projects.

- **Sample**: preparing the data set for the purpose of modelling through data sampling.
- **Explore**: evaluation of the data prepared at the previous step, looking for correlations and relations between variables to find patterns, normal and abnormal behaviours in the data sets, this step usually is performed via visualization.
- **Modify**: this step is associated with renovation, creation and modification of the variables in order to prepare modelling.
- **Model**: applies data mining and data modelling techniques as well as prepared variables for producing models.
- **Assess**: inspecting the accuracy and reliability of the produced model.

Data mining techniques have demonstrated limited capability in real projects involving multiple types of processes.

Data mining utilizes techniques to mine data and to discover and extract abstract patterns from the data sets and show them in different formats like rules or decision trees. Those patterns are applied for extracting knowledge like data groups, abnormal records and dependencies. Data mining is not dealing with details and does not give deep insights about the processes. Interpretation and result reporting are out of the scope of data mining.

2.4.2. Process mining

Process mining combines data mining and process modelling to generate models from IT log data. The produced models can be updated fast so they mirror real-life situations better than data mining, and hence can be used for many (additional) purposes. Examples include discovery of bottlenecks, analysis of social networks, work balancing, detection and prediction of (dis)similarities and/or flaws, etc .Nowadays a large number of factories are implementing process mining techniques to cope with problems that are not recognisable by common data mining techniques or by tools used for online monitoring of the systems.

The start point for process mining is an event log. The fundamental plan is to extract knowledge from event logs recorded by an information system. Process mining aims at discovering, controlling and improving processes by generating models explaining the behaviour of systems and analyzing them[28],[29]. There are three types of process mining projects[11]:

- *Data-driven* (also referred to as "curiosity driven"), i.e. powered by the availability of event data. There is no concrete question or goal, but in contrast with data mining, data-driven process mining projects are more process centric than data-centric. Process mining looks at the data from process aspect so that each process consists of a set of events and each event is demonstrating one executed activity. Process mining usually focuses on those parts or processes which are in concern.
- *Question-driven,* aiming to answer specific questions, e.g., "Why do cases handled by team X take longer than cases handled by team Y?" or "Why are there more deviations in weekends?".
- *Goal-driven,* i.e. aspiring to improve a process with respect to particular Key Performance Indicators (KPI) such as cost or response times.

Process mining techniques can be categorized as follows:

First, the *process (control-flow) perspective* is responsible for activity flow control, i.e., the order of activities. In this perspective the main goal is to discover and characterize all possible paths and exhibit them in process models such as Petri nets, EPCs, BPMN and etc. Examples include, but are not limited to exhibiting feasible paths on Petri Nets or event-driven process chain (EPC).

Second, the *organizational perspective* presents information about the concealed resources existing in the log, i.e., performers or actors like people, systems, roles, departments and their relations (e.g. a social network).

Third, the *case perspective* evaluates specific cases based on their properties. Each case can be characterized according to their path in the process model or based on the actors performing on the case. For instance, if a client shows high interest about a product, it will be interesting to recognise the supplier and the number of product each time they order.

Fourth, the *time perspective* takes into consideration execution time and frequency of events. In this perspective it is necessary for events to contain timestamp. In this perspective bottleneck discovery, operation of resources or predicting the remaining time of processes is mostly considered.

Based on the availability of a prior model, process mining techniques are divided into three different classes (*Figure 5*):

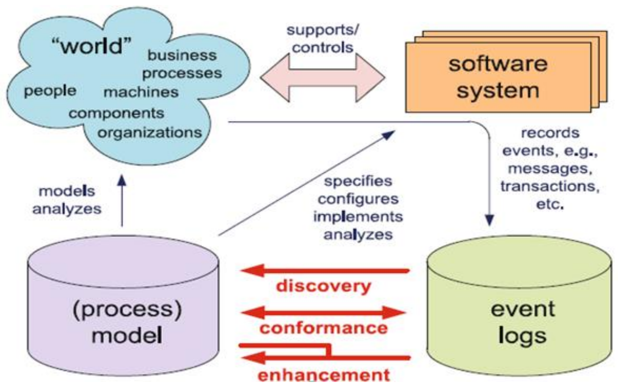


Figure 5. *Process mining overview [11]*

1. Process mining for *discovery*; here a model representing the causal dependencies between activities is produced from an event log without any prior knowledge about the system. There are different techniques for generating model from event logs. Modelling languages like Petri nets, BPMN, EPCs and ULM Ads have shown complexities in order to find and represent observed behaviours in a transparent and concise way. Process discovery is demanding, and the discovered models are inclined towards deadlocks and live locks. Four criteria are introduced to validate process discovery [11]: First, *fitness* refers to the ability of a model to replay majority of the events and traces in the event log. Second, *precision* quantifies how far the discovered model behaviour is from the behaviours recorded in the event log. Third, *generalization* **(**a model not complying with the generalization criteria can be referred to overfitted in the framework of data mining). Fourth, *simplicity*: the discovered model should represent the behaviour in an easy-to –understand way and as simple as possible (Occam's Razor).
2. Process mining for *conformance*; here an available model of the system is compared with the event log of the same system. The aim of the comparison is to recognise whether the reality is equivalent to the model. Conformance analysis can be applied for detecting deviations. For example, a decision miner described in [48] which takes a priori process model and analyzes every choice in the process model. For each choice the event log is consulted

to see which information is typically available the moment the choice is made. Then classical data mining techniques are used to see which da*ta elem*ents influence the choice. As a result, a decision tree is generated for each choice in the process.

3. P*rocess mining* for *extension***,** i.e. repairing the model for the purpose of better reflecting the reality or adding new *properties to* **t**he model by evaluating the correspondence between model and logs. e.g., bottlenecks are shown by colouring parts of the process *model.*

*P*ossible steps in a process mining implementation (*Figure 6*) include [53]:

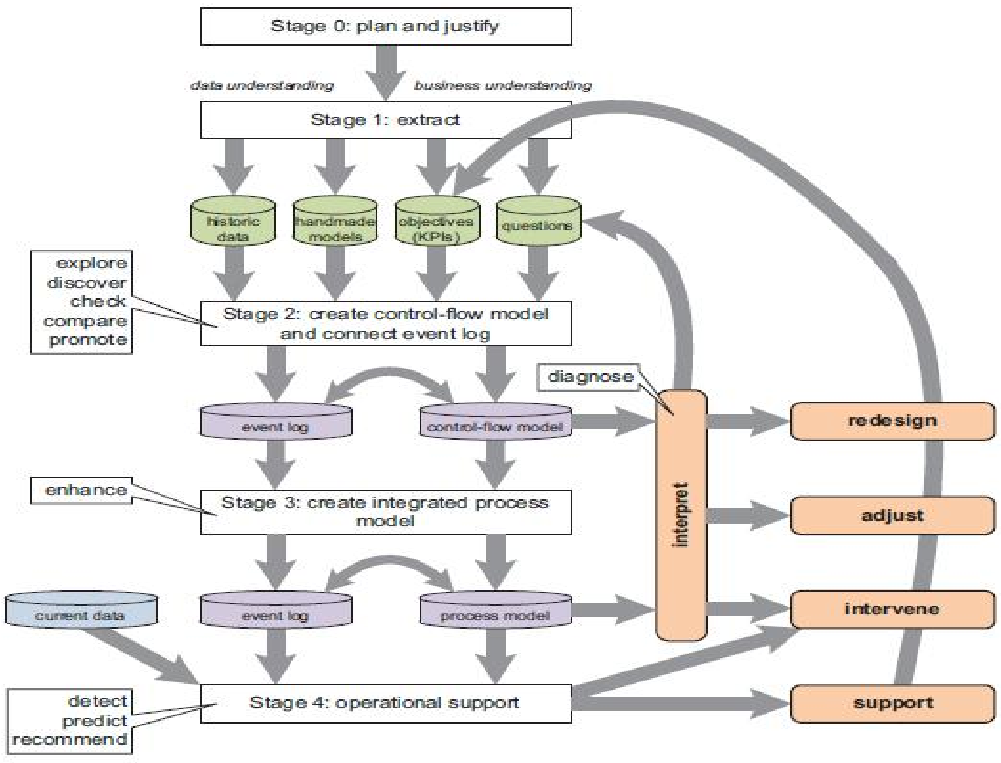


Figure 6. *Concrete activities in process mining life cycle [53]*

Stage 1: Planning, designing and verification of ideas / desired type of process mining project. This stage focuses on questions to be answered, goals to be planned and benefits to be forecasted. Historical data, models, KPIs and questions from all possible resources are pulled out based on the plans generated.

Stage 2: Generating the control-flow model and associating it with the event logs. At this stage, multiple automated techniques may be used for the purpose of control flow discovery (see *Table 3*). This stage is concerned also with the filtering of useless events or addition of new event types, and tuning of the model based on conformance checking or comparison of a predefined model with the discovered model. At this stage, the model is discarded if the fitness calculated is lower than 80% (this threshold varies depending on whether the underlying process is Lasagna or Spaghetti (2.3.2)). The model tightly linked with the event log must be interpreted before any redesign, intervention or adjustment (see *Figure 6*).

Stage 3: The model is added perspective(s) to make it useful for many different goals. An approach integrating the model with time, case and organizational perspectives is presented by Van Der Aalst [11].

Stage 4: Operational support (including prediction, detection and recommendation) based on current data. Interpretation of results is no longer needed at this stage: for instance, automatic emails concerning abnormal behaviours can be sent to the responsible personnel. Preconditions for this stage include an input structured process like Lasagna. And a high quality of event logs [53], [29]**.**

Many different technologies support business intelligence. Business Activity Monitoring (BAM) is a technology supports real-time monitoring of business processing. Complex Event Processing (CEP) is the technology of processing large amounts of events in order to monitor, direct and optimize the real time business. Corporate Performance Management (CPM) analyzes the performance of the processes. Managerial approaches such as Continuous Process Improvement (CPI), Business Process Improvement (BPI), Total Quality Management (TQM), and Six Sigma are technologies analyzing the processes deeply for the purpose of discovering rooms for improvements. Almost all of the mentioned business intelligence tools and management techniques and technologies can be extracted from the capabilities of process mining techniques. Great interest in process mining is observed from industrial side[49].

Some analysts limit the capabilities of process mining only to some *specific* data mining techniques.

Figure 7 shows the life cycle of Business Process Management (BPM).A model is designed or a predefined model is redesigned. In configuration/implementation phase the model is converted into processes executing within the systems. After that, the processes are executed and monitored and based on the knowledge taken from monitoring the process is adjusted. At the diagnosis stage, the process is evaluated and based on the demands and environmental effects, some changes might be done on the model or a new model might be designed.

Design and configuration are linked to the models, while monitoring and diagnostics are connected to the data. Advances in process mining made it possible to cover the entire BPM life-cycle. In addition to diagnostics, process mining now supports some operational aspects of the execution side, and the preparation of recommendations / predictions based on models extracted from historical information.

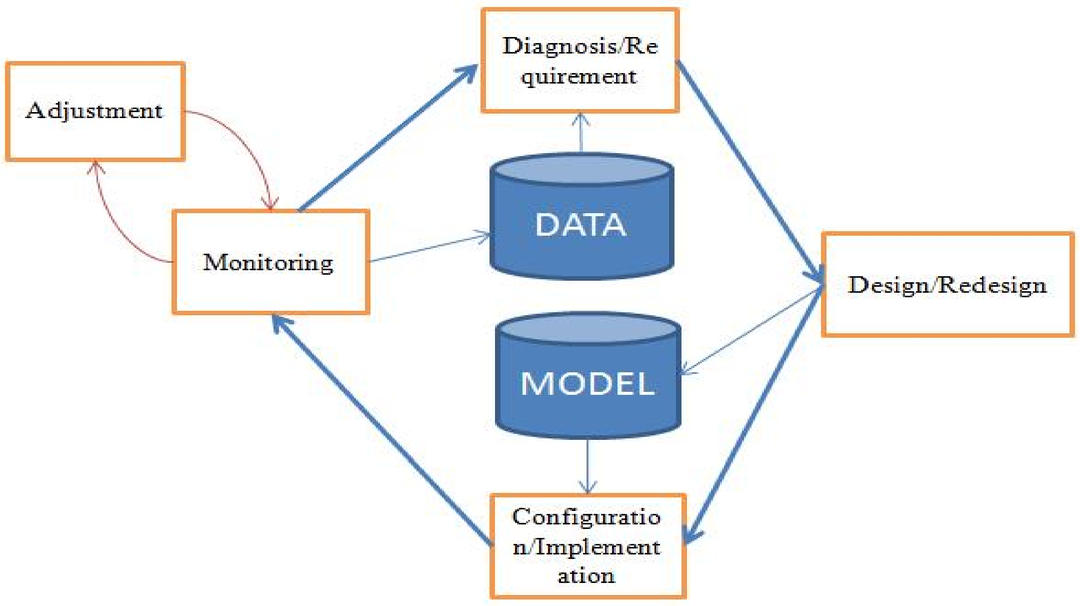


Figure 7. *BPM life-cycle[11]*

2.4.3. Data vs. process mining

Data driven approaches consider data from an outside perspective (frequency, average, service level and etc), and represent aggregated views of data. They are deployed by visualization tools. Detailed analysis of process performance is not easy to achieve by data driven performance analysis approaches. *Process driven approach* relies on a model of the system as a metaphor for the whole. The challenge here is the need to have an underlying accurate and well-organised model, which is difficult to impossible for complex processes.

Data mining looks at the data collection as a whole, and relies on the entire data to achieve decision making / provide recommendations. Process mining usually focuses on parts of processes which are of concern at a specific time.

Data mining is data-centric, which focuses on the raw data or data coming from condition monitoring. Process mining is process-centric, which looks at the batches of data or events (activities) as process instances and analyze them according to correlations inside processes or among them.

In both data mining and process mining generalization is an important subject. Data mining generalizes to avoid overfitting (neglecting outliers); process mining generalizes to cope with complex processes and just focus on the main process flows. Both techniques generalize in order to cope with noisy data.

With data mining, an increase in the complexity of the processes results in a serious decrease in accuracy of prediction. Popularity of process mining is because of its capability to produce structured models from Spaghetti processes.

2.5. Data Mining Application: Fault detection and diagnosis

A fault is defined as the deviation from an expected scope of calculated equipment variables / parameters. Fault detection inspects the irregularities of system to ensure successful operation. In [65], three steps are specified as the main steps of fault detection procedure: alarm, identification and evaluation.

Fault detection is a subset of control engineering. It is associated with monitoring of a system, searching for a fault and the time of its occurrence, identifying the location of the fault and categorising the type of the fault. Fault detection methods can be divided into model-based or data-based categories. Model-based fault detection requires a priori knowledge about the processes. This knowledge is provided by applying a mathematical model of the system used as a reference for analyzing the new sampled data. Increase in the complexity of the system leads to challenge of ensuring the accuracy of the extracted model. In contrast with the model-based fault detection methods, data-driven fault detection requires only the historical observations of the process performance recorded in database. In data-driven fault detection approach there is no need for any accurate mathematical or physical model of the system. Behaviour of systems is inspected using either statistical or non-statistical techniques. Data-based methods are applied when the physical model of the system is complicated or when the basic system operation principles are difficult to model but there is enough monitored data available concerning the system. Although data-based methods resolve most of the difficulties of applying model-based methods, they bring us to the main challenge associated with data-based fault detection, i.e. the need for large quantities of good quality training data.

According to [66], [67], extraction of features from datasets can be divided into qualitative and quantitative methods.

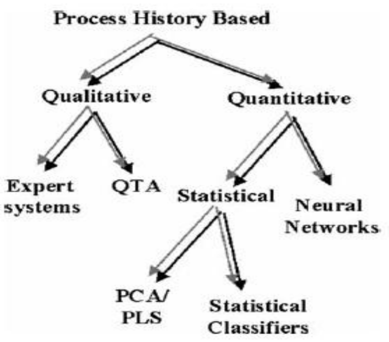


***Figure 8**. Classification of process history-based methods[66]*

Data-driven fault detection is based on historical observations of process data. The major methods for qualitative history information are expert systems and trend modelling methods. Figure 8 shows that quantitative information is classified into statistical and non-statistical algorithms. Principle Component Analysis (PCA), Partial Least Squares (PLS) and statistical pattern classifiers are the main methods applied on statistical quantitative feature extraction. Neural Networks is one of the most popular methods of non-statistical quantitative feature extraction and nowadays Support Vector Machines (SVM) has shown high capabilities and widely applied instead of NNs.

NNs are developed based on heuristic path with wide experiments. SVMs were developed based on theory, implementation and experimentation respectively. One of the most significant disadvantages of neural networks (failing to find the global optimum) appears when there are numerous local optimum values. SVMs always have a global and unique optimal solution. Another advantage of SVMs as opposed to NNs is “sparseness”, i.e. Classifiers are produced by SVMs only based on support vectors and not dealing with whole data. Last but not least, SVMs can deal with data sets including large number of features. Numerous features cause data sets to have high dimension. Since SVMs have the capability to apply kernel tricks, it makes them a significant and smart technique for high-dimensional data sets. ANNs result in overfitting when applied for regression or prediction purposes [102].

2.5.1. Support Vector Machines (SVM)

Vapnik (1995) introduced Support Vector Machines (SVM), as a data classifier and a nonlinear function estimation tool.

SVMs are generally two-class classifiers (binary classification) [67]. Multi-class classification is accomplished using combination of binary classifications and a decision making procedure. In order to have a classification on a dataset consisting of multiple labels (multi-class classification), the most used methods in practice are one-versus-all and one-versus-one classifications. One-versus-all SVM is performed for each class by disguising between that class and all remained classes and applying winner-takes-all as decision making strategy about the fitting class. In one-versus-one approach, classes are evaluated in pairs. One-versus-one decision making procedure is based on max-wins voting strategy. It defines a discriminated function in which the value of SVM for two classes, i.e. (c1, c2), is calculated. In the case of positive value the class c1 wins a vote and in the case of negative value the class c2 wins a vote. Finally the class with highest number of votes is assigned to the test pattern.

About binary classification, SVM is deployed to classify a dataset into two classes like {+1,-1}. The purpose of the SVM is to provide a hyperplane as a boundary between two classes of data. One of the features of SVMs is their capability to classify non-linearly separable data. Using kernel functions, SVM maps the input data to a higher-dimensional feature space where training set is linearly separable. *Figure 9* illustrates how projecting data into a higher dimensional space makes them linearly separable [67].

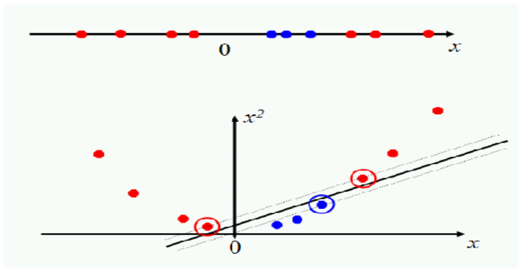


Figure 9*. Mapping non-linearly separable data into a higher dimensional space*

The most commonly used kernel functions by SVMs are linear kernel, polynomial kernel, multi-layer perceptron kernel and guassian radial bases kernel function.

2.5.2. Least Squares Support Vector Machines

One of the most interesting properties of SVM is sparseness, i.e. a large number of elements in convex quadratic programming (QP) problem are zero. However, for data sets with large amount of data, SVMs have shown to be time and memory consuming from optimization point of view. This problem is solved by introducing LS-SVM [68] which solves linear equations instead of QP problems. Although LS-SVM results in easier-to-solve equations, they suffer from lack of sparseness. However, this problem is also overcome in [69] so that a *simplest pruning method* is defined in the context of LS-SVM. After all, LS-SVM is preferable in large scale problems and by applying *pruning* method, sparseness is solved and the performance is similar to SVM.

The major difference between SVM and LS-SVM is that SVM solves a burdensome quadratic program problem for training while LS-SVM overcomes that by solving some linear equations [69].

2.6. Process Mining Applications

2.6.1. Process Mining for Discovery

Stage 2 of process mining (Figure 6) is the Discovery of a model based on historical data.

Given an event log L, a process discovery algorithm is defined as the function outputting a representative model of the input log. Table 3 introduces several control-flow discovery algorithms in the context of process mining. Depending on the goals, the analyzer selects an algorithm.

The following subsections give a brief theoretical overview of two approaches to discovery: the alpha algorithm and respectively the heuristic approach.

2.6.1.1 Discovery based on alpha algorithm

The alpha algorithm is a simple algorithm whose principles have been embedded in other powerful algorithms.

Figure 8 illustrates the basic steps required to produce a Petri net model by α-algorithm from an event log.

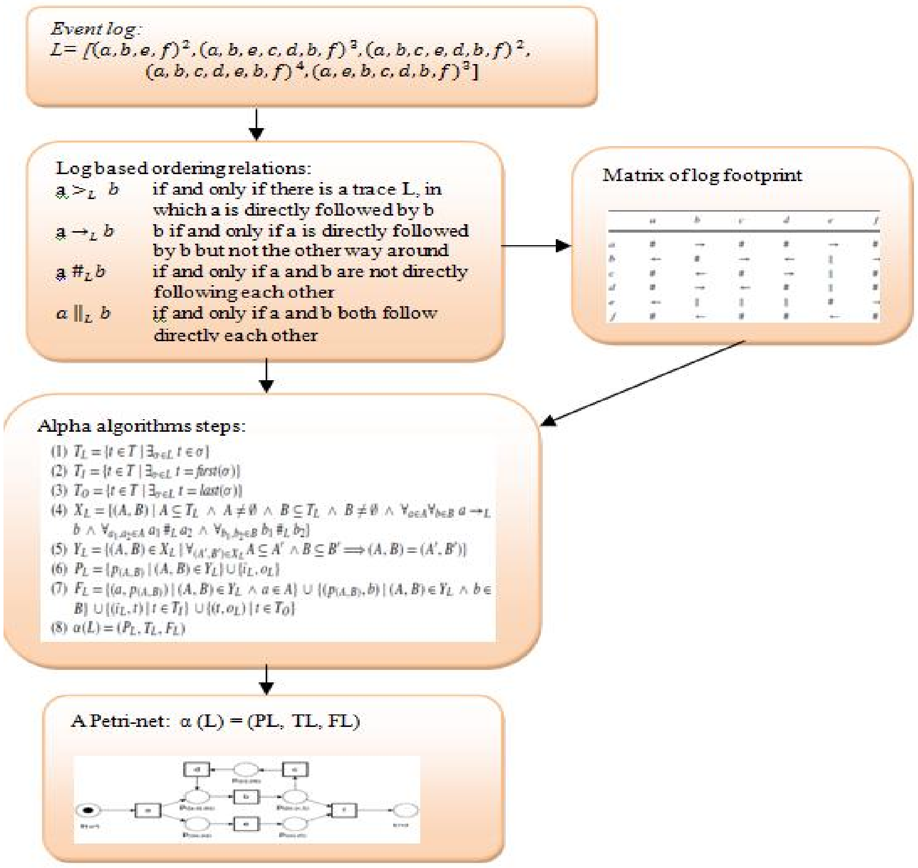

Event log:
L= [(a,b,e,f)², (a,b,e,c,d,b,f)³, (a,b,c,e,d,b,f)², (a,b,c,d,e,b,f)⁴, (a,e,b,c,d,b,f)³]
Log based ordering relations:
a >L b if and only if there is a trace L, in which a is directly followed by b
a →L b b if and only if a is directly followed by b but not the other way around
a #L b if and only if a and b are not directly following each other
a ∥L b if and only if a and b both follow directly each other
Matrix of log footprint
Alpha algorithms steps:
(1) TL = {t ∈ T | ∃σ∈L t ∈ σ}
(2) TI = {t ∈ T | ∃σ∈L t = first(σ)}
(3) TO = {t ∈ T | ∃σ∈L t = last(σ)}
(4) XL = {(A,B) | A ⊆ TL ∧ A ≠ ∅ ∧ B ⊆ TL ∧ B ≠ ∅ ∧ ∀a∈A∀b∈B a →L b ∧ ∀a1,a2∈A a1 #L a2 ∧ ∀b1,b2∈B b1 #L b2}
(5) YL = {(A,B) ∈ XL | ∀(A',B')∈XL A ⊆ A' ∧ B ⊆ B' ⟹ (A,B) = (A',B')}
(6) PL = {p(A,B) | (A,B) ∈ YL} ∪ {iL, oL}
(7) FL = {(a, p(A,B)) | (A,B) ∈ YL ∧ a ∈ A} ∪ {(p(A,B), b) | (A,B) ∈ YL ∧ b ∈ B} ∪ {(iL,t) | t ∈ TI} ∪ {(t,oL) | t ∈ TO}
(8) α(L) = (PL, TL, FL)
A Petri-net: α (L) = (PL, TL, FL)

Figure 10. *Amodel representing steps to produce a work-flow model by alpha-algorithm*

The sample event log includes 6 distinct activities (a,b,c,d,e,f) combined into 14 (2+3+2+4+3) traces among 5 cases.

Based on 4 log-based ordering relations, relevant patterns inside the log are recognised and a matrix (log footprint) including all the relations based on the defined rules is produced.

The 8 steps for *α-algorithm* are*:*

1. Finding activities inside the traces , T_L = {a, b, c, d, e }
2. First event of each trace , T_I= {a}
3. Last event of each trace , T_I= {f }
4. The purpose is to find places p (A,B). A consists of the set of input transitions and B consists of the set of output transitions so that elements of A (a,b,..) should follow the rule; a $\#_L$ b and the same for B. To clarify, A consists of all elements having causal dependencies with all elements of B, however there

should not be any dependency among the elements of A or B. For instance, given log L1; A={a,d} , B={b} are satisfying the rules explained above. There are transitions from a to b and from d to b, but no dependency between a, d. At this step XL including set of mentioned pairs is produced.
X_L= {({a},{b}), ({a},{e}), ({b},{c}),({b},{f}),({c},{d}),({d},{b}),({e},{f}), ({a, d},{b}),({b},{c, f})}

5. In order to reduce the number of possible places, YL as a subset of XL excluding non-maximal pairs is produced.
   Y_L= {({a},{e}), ({c},{d}), ({e},{f}), ({a, d},{b}), ({b},{c, f})}
6. P_L Consists of places produced based on elements of Y_L. Considering pairs defined in (A,B) ϵY_LYL , each pair is denoted as a place P(A,B) connecting transitions of A to transitions of B. It also includes source and sink places.
   P_L= {P({a},{e}), P({c},{d}), P({e},{f}), P({a, d},{b}), P({b},{c, f})}
7. F_L including transition of pertinent is produced.
   F_L= {(a, P({a},{e})),(P({a},{e}), e), (c, P({c},{d})),(P({c},{d}), d), (e, P({e},{f})), (P({e},{f}), f), (a, P({a, d},{b})),(d,(P({a, d},{b})), (P({a, d},{b}, b), (b, P({b},{c, f})), (P({b},{c, f}), c), (P({b},{c, f}),f), (start, a) , (f, end)}
8. A Petri net model by applying α-algorithm on the event log L is generated. The output of α-algorithm contains, places, transitions and events (activities)

2.6.1.2 Discovery based on the heuristic approach

The α-algorithm has shown weakness in dealing with the four validation criteria identified for process discovery (fitness, precision, generalization and simplicity, see Section 2.4.2. It is a rather primitive discovery method, as it has shown problems in practice, e.g. noise (less frequent behaviours), incompleteness (i.e. failing to include all relevant traces in the resulting log) and complex routing constructs.

The output of heuristic mining approach is a model similar to C-net. The algorithm is robust because of the representational capabilities of causal nets. Specifically, C-nets take into consideration the number of times that each activity occurs in a log (its frequency), and associates it with log-based ordering relations.

For instance, the absolute value of a $>_L$b (Figure 8) is the number of times activity a is followed by b and represented by $|a >_L b|$.

The notation a$\rightarrow_L b$ refers to the number of times a is followed by b but not the other way around.

The notation $|a \rightarrow_L b|$ refers to the dependency relation between a and b:

$$|a \rightarrow_L b| = \begin{cases} \dfrac{|a >_L b| - |b >_L a|}{|a >_L b| + |b >_L a| + 1} & if\ a \neq b \\ \dfrac{|a >_L a|}{|a >_L a| + 1} & if\ a = b \end{cases} \tag{1}$$

In case a ≠ b , the output of dependency relation will be a number between 1 (i.e. *a* is often followed by *b*) and -1 (i.e. *b* is often followed by *a*). A dependency value close to zero, declares both *a* and *b* follow each other.

In case of loops, like when activity *a* occurs sequentially, a=b , the formula for measuring dependency is modified to the second condition of (2). After computing the dependency between all the activities of an event log, a matrix including all dependency values is produced. Having dependency and frequency metrics, a dependency graph can be generated. Based on the thresholds defined for dependency and frequency, activities are appeared in the graph. For instance, defining the threshold 2 for frequency causes those activities which have happened less than two times to not be appeared in the graph and the same for dependency threshold.

As mentioned before, the output of a heuristic approach is a C-net model, C = (A,a_i, a_0, D, I, O). The nodes of *dependency graph* stands for parameter A and the arcs of *dependency graph* stand for parameter D in C-net model. The C-net graph, considers a start (a_i)and an end (a_0) activity for all the traces so that traces without those unique start and end events should not be considered. Therefore, dependency graph explained above is the core of C-net graph. Defining functions I and O completes the C-net from dependency graph. For instance, if ao = {b,c,d,e} which means that output of activity a is activities b,c,d,e. Then, O(a)= 2^4-1=15 and there are 15 possible output bindings for activity a. If oa={f,g}, which means f,g are inputs for activity *a*, then I(a)= $2^2 - 1$=3 , which means that 3 possible input bindings exist for *a*. If one activity contains only one potential binding element, C-net considers that activity in the model. Since all the potential bindings are not in the event log, by replaying the event log on the dependency graph, the occurrence of input and output bindings are recognised. Then, we can define some thresholds to remove those bindings which are less frequent. Finally, functions I and O are obtained and C-net is modelled from *dependency graph*.

The strong point of the Heuristic miner is its robustness about noise. Having the possibility to define threshold for dependency and frequency makes it possible to only extract useful or the core of data instead of modelling details of information.

2.6.2. Basic Performance Analysis

Usually, increasing the load of a system leads to decrease in performance of the system. Scalability of a system is defined as the capacity of the system in response to higher loads. Modifying the scalability of a system in order to accept and respond to higher loads is called *performance tuning*. Five steps can be defined for performance tuning: providing values projecting the behaviour of the system, calculating the performance of the system before modification, focusing on part of the system which is crucial for the performance enhancement point of view (bottleneck), modifying those parts counted as bottlenecks and measuring the performance of the system again in order to evaluate the progress in performance.

Performance analysis discloses different aspects of a problem or opportunities and recognises all the drivers or obstacles to make a performance successful. Usually performance analysis proposes solutions based on discovery. It has a wide scope, but it can be divided into three dimensions: time, cost [56] and quality (mostly related to the product or services after delivery). As evaluation of performance analysis is firmly connected with KPIs, for each dimension different performance indicators is defined.

Time dimension, for instance, may be assigned one to all of the following indicators:

- Lead time: the overall time of the case to be completely performed. In this dimension, service level is defined as the percentage of the cases that overall performance time of them is lower than a predefined threshold. This KPI measures the average of time throughout the case, accounting for variance.
- Service time defined as total time takes for one case or one activity of a case. In fact, service time is portion of lead time.
- Waiting time: the time a case or activity waits for a service. Moreover, service level here defined as the percentage of the activities waiting for a service during the time defined by averages of time intervals.
- Synchronization time: the time an activity waits for another transition to be triggered.

Process mining and performance analysis

Some of the common questions raised at stage 1 of process mining are: “how is the real process working?”, “where are the bottlenecks?”, “how and with whom the tasks are performed?”, “how is the communication between components?”, “looking at the event logs, is the system working as organizations expect?”

Process mining generates a broad overview of many different aspects of the same process: model representations of throughput times, the control flow of the process[57],[58], [59], process conformance[60], overview concerning the communications (social-network [61],[62].

Most of the process mining works have been done by researchers having enough and proper knowledge about the system. However, complex information systems make it difficult for different organizations to interpret results with less prior knowledge about the system. At factory floor, in addition to the overview about the whole performance of the process, usually process mining is carried on in order to generate diagnostics about the parts which are malfunctioning. But those parts are not recognisable by common data mining techniques. Process mining produces numerous insights relying solely onevent logs.

According to [63], the performance of a system is explained as the response time of a system to the requests. They also define criterion to evaluate failure or success of a process as the number of the requests that a system can process at less than a defined period of time. Considering the bottleneck definition discussed before, crucial part of a system for the performance enhancement point of view, bottlenecks are recognised as the main obstacles on performances. Discovering the place of a bottleneck during testing hardware under load is not a hard task because the hardware is usually monitored easily while operating. In contrast, discovering the bottlenecks at software side or logical supports like task instances or buffers is not an easy task because these logical parts are not usually monitored or tracking them needs to cope with a substantial stream of data. At such complex cases, a process model can soften the problem. Having a model, evaluating both hardware and logical part of the process is less complicated. Hence using a model showing the performance of the system, there is no need to deal with a large number of parameters to discover the bottlenecks. It provides the analysts to just focus on those tests related to target bottlenecks. Although performance models have a lot of advantages, generating them as a model reflecting the system's behaviour with acceptable accuracy and conformance needs a lot of skills.

2.6.3. Process mining in practice: four related studies

Production automation systems rely on integrated engineering and cooperative information systems. In order to evaluate process performance, the challenge is to share the knowledge of different stakeholders from different worlds (engineering and business), which is usually represented via different models.

Process analysis is applied to analyze performance of multi-layered (business to process to machine) information systems for *industrial assembly workshops*[16] (*Figure 11*)At business layer, designed schedules and product specifications are input as a tree model to *conformance checking* against a process model derived from data/events collected directly from event logs generated by a simulator at the process layer. Failures based on the type of associated components (conveyors and robots) are classified into four classes (C1 to C4), and the relationship between these failures and the number of machine breakdowns is evaluated.

Finally organizational mining is performed. A model is produced representing structure of units for two purposes.

1) To show the components and their connections.
2) To show the social network of units e.g. how different units are cooperating.

Advantages of applying incorporated systems capturing events through different layers are presented. Flows of information between organizational units are demonstrated. Hidden information related to running processes is discovered. The work represents the classified failures, interaction of similar and different machines and output of systems using process inspection and organizational mining techniques.

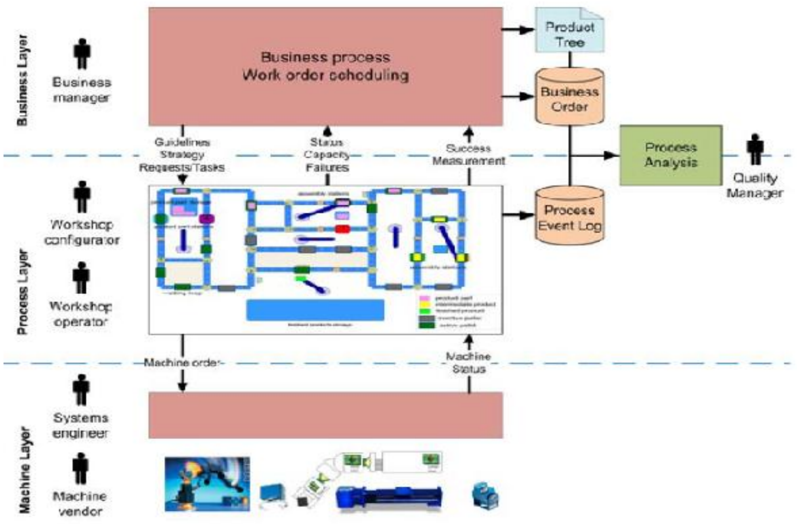


Figure 11. *Production automaton system structure*

ASML (the leading manufacturer of wafer scanners in the world) propose process mining approaches on *spaghetti-like* processes like ASML [71]and also show alternative mining techniques in order to provide a better insight of processes and more clear visualization. Three different categories of process mining are applied:

- *Discovery* on real processes, where a model is generated representing the frequency of the test operations on wafer scanners. The produced model gives a useful insight on how test processes are operating. The produced model is much less structured than the model predefined at design level (used as a reference model).
- *Conformance checking* between the produced model and the reference model is performed to search about how real processes are operating based on expectations.
- *Extension* of the discovered model by extracting extra knowledge from event logs.

The performance of the system is inspected for the purpose of discovering bottlenecks on the process model. Three perspectives (process / organization / case) of process mining are applied to analyze the management of invoices in the RWS office in the Netherlands[72], in order to discover the factors influencing the time of invoice payments. Different types of process mining techniques and tools beside multiple standards are combined to achieve the goal.

Process diagnostics as a mining method is discussed in [50]. A 5 step methodology is introduced: provision of event logs, inspection to find a general idea about the process events, analysis of the control work flow, performance analysis and analysis of the role of the different components.

Despite several successful research results on process analysis, there is still distance between theory and practice, because of dissimilarity in data sources and lack of qualified data collection [13],[14],[15].

2.6.4. Available toolkits

IBM Cognos Business Intelligence (IBM), Oracle Business Intelligence (Oracle), and SAP Business Objects (SAP) are the most applied Business Intelligence (BI)-based software products presented by different vendors. According to [11], most of the BI-based software products are data-centric and they provide simple or less professional ways of analysis.

Process mining research started from mid-nineties. After growing of data mining, there were some attentions to processes. On that time the research area was mainly about process model discovery using event logs; however there was not enough quantity of event logs and process mining approaches were immature[1]. In 2002 some tools for the purpose of process mining were proposed; MiMo (α-miner based on ExSpect), EMiT (α-miner taking transactional information into account), Little Thumb (predecessor of the heuristic miner), InWolvE (miner based on stochastic activity graphs), and Process Miner (miner assuming structured models)[77], [11]. However they were not powerful enough to be applied in real life. During the last decade, as a result of availability and maturity of event logs, process mining techniques have shown impressive progress. In 2004 the first version of Prom (1.1) tool [77] has been developed at Eindhoven University of Technology by Wil van der Aalst and colleagues, and currently is available as an open source toolkit. Prom utilizes a plug-able architecture in which builders can design and apply their own process mining techniques. The main advantage of Prom for developers is that they will develop their own plug-in without making a lot of effort for loading or filtering of the event logs. Currently there are more than 280 algorithms implemented and accessible by user as plug-ins. The framework provides straight and easy user interface functionality, multiple ways for implementing different model types (e.g. Petri Nets, EPCs, etc.) and it also provides an easy and common way for reading files. The input of Prom is event logs in MXML format and from ProM version 6 the log format XES [18], can be used. Moreover, Prom offers easy-to-use applications for visualization of the results [71].

Listing all tools supporting process mining is an impossible task.

Table 2 shows some examples of both commercial and academic available products with process mining abilities.

Table 2

Process mining tools, short overview

	Product	**description**
1	ARIS Process Performance Manager(SOftware AG) [85]	A tool for Process Intelligence Solution, provides assessment of the processes considering speed, cost, quality and quantity, and identifies optimization opportunities [85]. Does not support conformance checking, prediction, and recommendation.
2	QPR ProcessAnalyzer[86]	Automated Business Process Discovery.
3	Disco[87]	A complete process mining tool Proposed by Fluxicon. High capability to deal with Spaghetti- like processes.
4	Celonis Discovery [88]	Reconstruction and analysis of ERP processes
5	Celonis Orchestra	Monitoring, analysis and optimisayion of IT services.
6	Nitro	Proposed by Fluxicon, reads data from Comma Separated Values (CSV) or Excel files and converts to XES or MXML event log formats [90].
7	Comprehend (Open Connect)	Provided by OpenConnect ,a pioneer in process intelligence and workforce analytics solutions. http://www.oc.com/technology/
8	Discovery Analyst (StereoLOGIC) [91].	Understanding and improvement of business processes in real time. Visualization and comparison of processes being executed
9	Flow (Fourspark) [92].	Can deal with large-scale industrial data with incomplete transaction logs, and inconsistent log structures
10	Reflect|one(Pallas Athena)	Mature commercial products and have wide process mining capabilities [93],[94]. Reflect does not support conformance checking and prediction but supports the entire BPM life cycle [11]. The user need not be an expert.
11	Reflect (Futura Process Intelligence)	
12	Interstage BPME	Interstage Automated Process Discovery, Fujitsu proposal as a commercial service (not a system to be installed) . Process discovery [95].
13	Enterprise Visualization Suite (Busines scape)	Concentrates on business processes which are provided and supported by SAP [96].

2.6.4.1 **ProM Framework Tool and relevant Plugins**

Currently ProM version 6 includes about 280 plug-ins which target various application domains. Tables 3 and 4 illustrate a short list of selected ProM plug-ins, relevant for this thesis, applicable for discovery and/or conformance check.

Table 3

ProM plugins applicable to process discovery

Output process model	**Plugin**	**description**
Petri net	Alpha algorithm	Discovers a Petri net using the α-algorithm [34]
	Alpha ++ algorithm	An alternative for α-algorithm in case of loops [35], [36].
	Parikh Language-based Region miner	Parikh miner tends to derive very aggressive abstractions [37].
	Tsinghua-alpha Algorithm plugin	An extension of alpha algorithm [38]
	Petrify miner	synthesizes a Petri net with a *reachability* graph that is *bisimilar* to the transition system [39], the synthesize algorithm is based on *theory of regions*[39].
EPC, Petrinet, Fuzzy	Conversion plug-in	Converts models produced by discovery algorithms to other model types, mainly Petri Nets
C- net	Genetic algorithm plugin	Discovers a C-net using genetic mining, Basic GA algorithm which is able to mine all the construct except of duplicate tasks [40].
	Duplicate Tasks GA plugin	An extension for basic GA, is able to discover also duplicate tasks.
	Heuristic miner	Represents a C-net using heuristic mining approach [41],[42].
Fuzzy model	Fuzzy miner	Discovering and representing a fuzzy model by applying fuzzy logic algorithm [43].
	Frequency Abstraction Miner	The importance of events and transition is evaluated by *frequency*, i.e. more frequently observed patterns are considered more important. Clusters of events organized according to frequency.
Finite state machine / Transition system	FSM miner	Having a number of settings, a log is mined for an accept finite state machine [44].
Event-driven Process Chain	Multi-phase Macro Plugin	Integration of multiple instance graphs into one EPC [45].
Other	DWS mining plug-in	Given an event log, this Plug-in builds a tree of work flow models. It discovers a set of characteristics (discriminant rules) in the log.
	WorkFlow Pattern miner	It discovers the local patterns instead of global ones [46][47].
	Doted chart analysis	Display of the distribution of events over time and computation of performance metrics. No reference or fundamental model is required. [72] .
	Performance analysis with Petri net	Bottleneck discovery by measuring the idle time between activities. Applicable for extension, when the model has high fitness value.
	Log summary tool	Overview of processes on the event log. Some general ideas like the number of cases or activities in each log and the sequence and throughput time of cases.
	Basic performance analysis	Calculates performance measures such as execution (working) time, waiting time, etc. Visual display.

Table 4

ProM plugins applicable to check conformance

Plug-in	**Description**
Conformance checker	Checking the conformance to evaluate how much the real process complies with the plans, by replaying the log on the reference or discovered model. Provides model and log views.
LTL checker	Applied for conformance checking. The difference between LTL checker and conformance checker is that LTL checker reports the discrepancies based on the rules which are defined, not based on the reference model.

There are some other tools which cannot cover all area of process mining but in conjunction with Prom they may be considered as a full capable product in the context of process mining. For instance, *Genet*, *Petrify*, *Rbminer*, *and Dbminer*[97] are supporting only process discovery capabilities and for conformance checking techniques they rely on ProM.

3. TESTBED

The testbed used for this research work (henceforth denoted as FASTory) was previously used in a real factory for assembly of mobile phone components. Figure 12, illustrates the layout of the line.

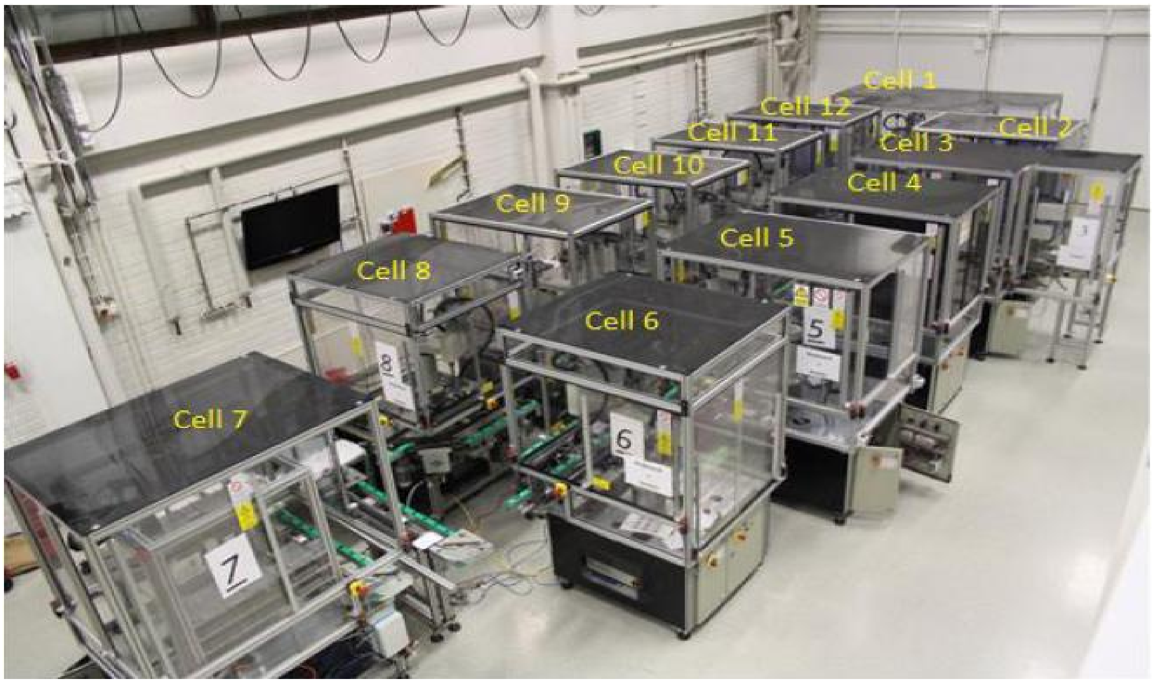


***Figure 12**. FASTory Line*

FASTory line is a pallet-based production line consisting of 12 cells. Ten out of twelve cells are workstations. Each workstation includes one main conveyor, one bypass conveyor and one SCARA robot (SONY SRX-611). As shown in *Figure 13*, each cell is equipped with acrylic door and for safety point of view multiple interlock door switches and emergency buttons have been installed.

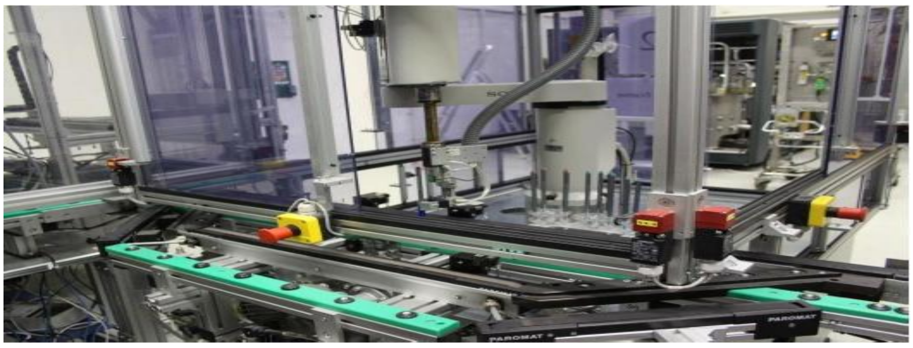


***Figure 13.** A detailed view of one cell of FASTory line*

Cell number one is loading/unloading station including a SCARA robot with a suction cap as end-effector. Using suction cap papers are loaded / unloaded on / from the pallets. Cell number seven includes only a simple conveyor as the connection, later this cell will work as a buffer in the chain of workstations. Raw products (drawing papers) are carried and transferred by pallets moving among the cells through the conveyors. When all ingredients of a recipe (operations should be performed on each paper) are finished (product carried by pallet is ready), the paper is unloaded at cell number 1 and a new paper is loaded on the pallet.

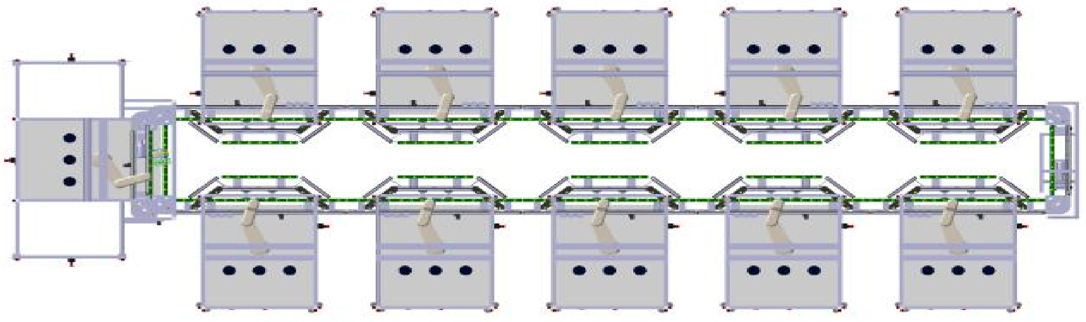


Figure 14*. Layout of FASTory line from top view*

The conveyor system of each cell (*Figure 13*) consists of one bypass (capacity of one pallet) and one main conveyor (capacity of two pallets). The operation of stoppers and conveyor belt is important for fault detection point of view which is explained in next chapter. *Figure 15* shows that there are 4 stoppers on each conveyor belt. One is located at the entrance, two on the main conveyor under the robot and one on the bypass conveyor. Each stopper is a pneumatic cylinder, by receiving a signal the stopper is activated and the cylinder is pushed into an aperture designed on the pallet and stops pallet from movement. Conveyor belt is rotated by conveyor engine continuously. Even when the pallet is stopped by the stopper, conveyor belt keeps turning. NFC readers installed beside each stopper, collect information regarding completed operations from the NFC tags carried by the pallets.

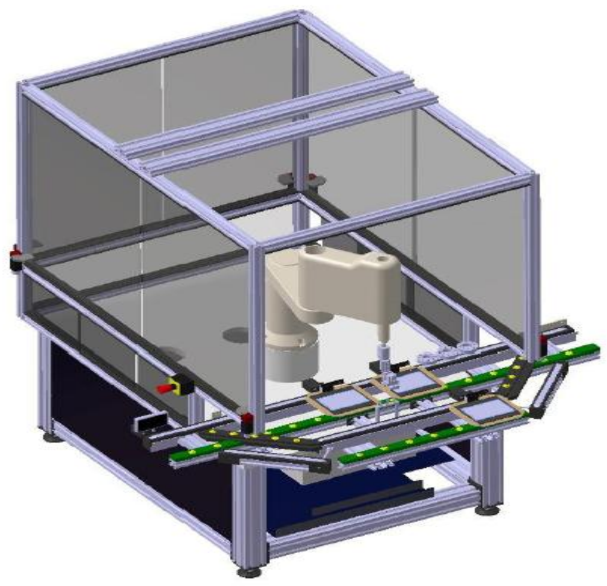


Figure 15. *Conveyor system analyzed. Main conveyor hosting 2 pallets. Bypass hosting 1 pallet.*

The initial version of FASTORY line was consisted of five robotic modular workstations previously applied in a factory for the purpose of assembling mobile phone components. The line was capable of producing one mobile type at each time. Entire sensors and actuators were connected via DeviceNet nodes to an OMRON PLC. Applied PLC was controlling processes of workstations in a centralized mode. An Ethernet network provided the communication of workstation controllers through coaxial cable following OMRON FINS protocol [78]. Each pallet has an RFID tag storing information about the operations completed on the product. Top of each stopper, an RFID reader has been installed. They read and send the information stored in RFID tags to the controller. For instance, using information provided by RFID readers, the position of each pallet in the line is identified by the controllers. RFID readers are connected over a DeviceNet interface. This interface provides a network between PLC controllers and I/O nodes, RFID nodes. According to *Figure 16*, protocols applied for the older version of FASTORY line are DeviceNet (getting and setting of the sensors and actuators status), RS232 serial communication applied for direct communication of the PLC controller and robot controller and Ethernet protocols providing a network between workstation controllers and manufacturing execution systems located at higher level.

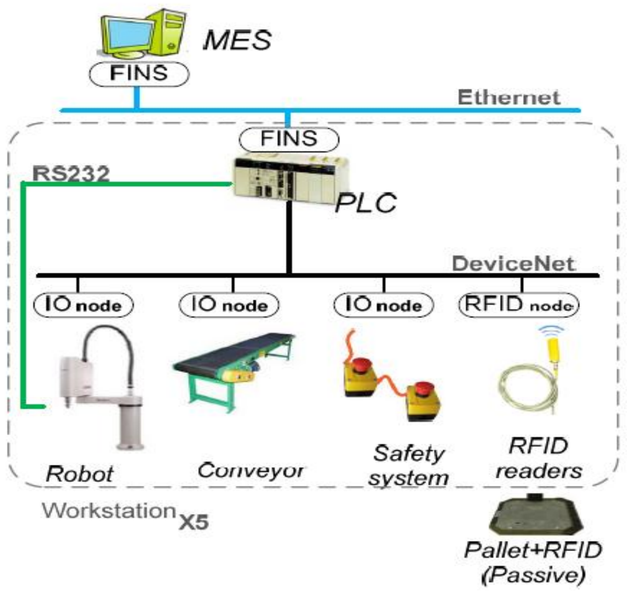


***Figure 16**. The previous system architecture for FASTORY line[101]*

FASTORY line was retrofitted on 2010 for the purposes like; higher flexibility, reconfigurability and asset-awareness, i.e. energy. The product output of the testbed was changed from assambled mobile components to painted mobile phone components (Frame, Keyboard and Screen) for research purposes. Each drawing operation can be deduced as an assembly operation. RFID tags/readers were replaced by NFC tags/readers. Web services (WS) based on Service Oriented architecture applied at device level and provides interoperation and integration of equipments and applications. Device Profile for Web Services (DPWS) stack paves the way for discovery and invocation of devices via web services [1]. The controller applied for all embedded devices is a commercial product called S1000 controller. S1000 implements DPWS stack to provide implementation of Web Services [80]. S1000 controllers are used as interface for sensors and actuators in FASTORY line supplying the communication of physical layer components among themselves or with external world. They are equipped with I/O terminals, RS232 serial port, Modbus/TCP or analogue inputs. S1000 controllers can communicate with each other or with upper layers by applying WS messages. They support both IPV6 and IPV4 network protocols. All components of the physical layer are coupled with smart RTUs (S1000 controllers) and communicating with all layers through web services. To clarify the subject, at the previous version of FASTORY line all embedded devices were exposed by nodes through DeviceNet, while in retrofitted version of the line all nodes are replaced with smart RTUs.

Figure 17 demonstrates the retrofitted architecture of FASTORY line. It is visible that all the components (robots, conveyors, pen feeders, NFC readers and etc.) are revealed as individual functional units controlled, discovered and subscribed independently via smart RTUs. There is no hierarchical structure like previous version i.e. devices at device layer were connected through fieldbus network protocols, while at higher layer the communication between controllers or Manufacturing Execution Systems (MES) was based on Ethernet which is another type of network protocol.

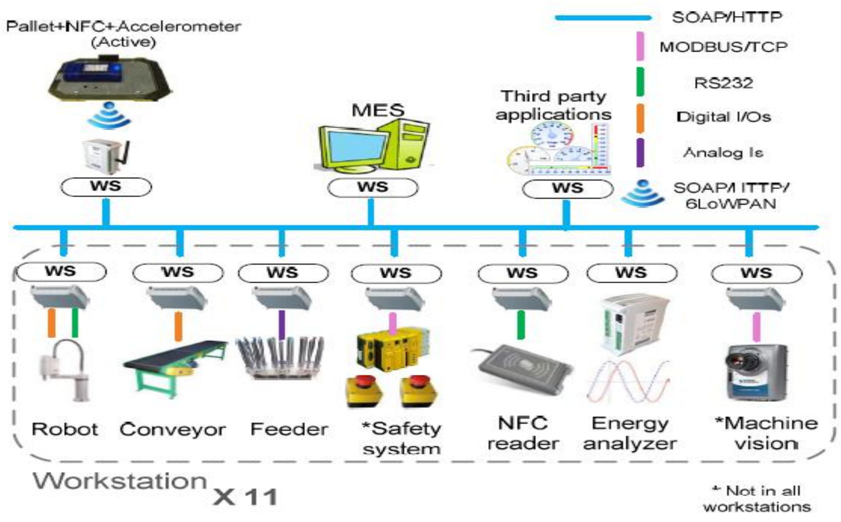


***Figure 17**. Architecture of retrofitted FASTORY line [101]*

The line is coordinated by a hybrid methodology, using client-server and peer to peer paradigms. The system is composed of physical devices in the line and a Decision Support System (DSS) located in an external computer. The devices and the DSS functionality are exposed as Web Services. The devices can communicate in a peer to peer fashion and with the DSS. All invocations and notifications are Web Service based. Exposed event notifications (Table 5) include information about energy consumption (via S1000 energy meters), CAMX state events (e.g. pallet input to a conveyor piece), quality, and temperature / humidity / light.

Table 5

Testbed generated messages

No	Message	Description
1	EquipmentChangeState	cell ID, recipe number, device type,pallet ID, the current state, theprevious robot state, time stamp.
2	QualityInspection	quality information including pallet ID, the quality of frame, screen and keyboard, the quality of the inspection result and a time stamp.
3	EnergyMeter	Robot/conveyor/controller energy consumption, per each working cell published at a time interval of five seconds.

Every time a pallet arrives to the entry point of a work station, the following events happen sequentially:
The NFC reader tag of the pallet is read. The conveyor invokes a routing decision in the DSS. The DSS checks the request and decides if the pallet will be processed or bypassed. This depends on the pallet, its status, and the work station capabilities. The

DSS responds the routing action. In case the pallet will be processed in the workstation, it sends the operation parameters (drawing component and colour). After this, the conveyor is responsible of routing the pallet until the next working station.

The devices also communicate peer to peer. Once the routing decision has been issued, the conveyor has the control over the pallet. If the pallet has to be processed in the working station, then there is peer to peer communication between the conveyor and robot. The conveyor requests the Drawing service from the robot. When the robot finishes its task it informs it to the conveyor and then the pallet continues its flow.

The product outputs of the testbed are drawn- mobile phone components. Each component (Frame, Keyboard and Screen) has three different types, and each type can be drawn by three different colours. Consequently, we have 9*9*9 =729 possible products as outputs. Given multiple types of the products, different control scenarios are considered for the line. Table 6 explains the scenarios:

Table 6

Possible control scenarios considered for FASTory line

	Scenario	**description**
1	Each cell can take all the operations at the same time	If the pallet bears raw paper and if the cell is empty pallet is transferred into the cell and robot do all the operations.
2	Each cell can perform only one operation at the same time	Cells are divided so that some of them perform only keyboard, some only frames and some only screens. Each cell is able to draw with every colour.
3	Each cell can perform only one operation with only one permanent colour at the same time	Similar to the previous scenario, but each cell is configured to draw only one colour
4	Each cell can perform all shapes of only one component	For example, each cell can draw only one shape like keyboard or screen, but including all 9 possible items for that shape such as colours and different forms of shapes.

All FASTory cells are equipped with energy meters integrated into S1000 processing units (smart Remote Terminal Units). Each energy meter is an E10 Energy Analyzer expansion module which provides 3-phase electrical power consumption monitoring (*Figure 18*). Phase A is consigned to the robot, phase B is allocated to the cabinet, I/Os and the controller and phase C is assigned to the conveyor system including the main and bypass conveyor. Power measurement is achieved by sampling current and voltage. *Figure 18* depicts the current sampled by a current transformer (CT) connected to +Ia-, +Ib- and +Ic- terminals and the voltage is measured by direct connection of the 3 phases and neutral to the Vn, Va, Vb and Vc terminals of the E10 expansion module. Equipment workload refers to the number of pallets occupying the conveyor at one time. To monitor this information, inductive sensors are mounted at entrance and exit points of each cell, by arriving or leaving each pallet, CAMX TransferIn/TransferOut notification messages are sent by controllers and then they are counted by a counter applied on the server side.

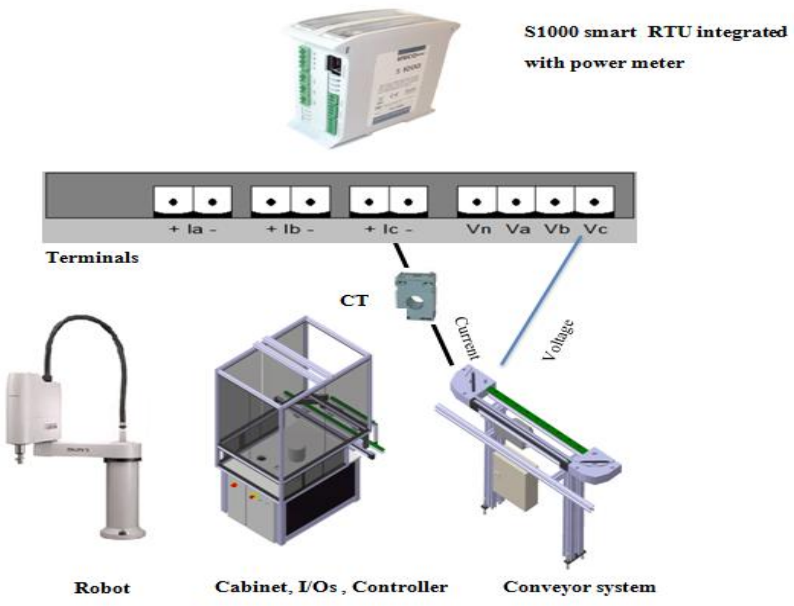


Figure 18. *Setting used to monitor energy consumption*

4. DATA AND PROCESS MINING IN PRACTICE: RESULTS

4.1. Data mining for fault detection and diagnosis: Incipient deterioration in transportation equipment segments

This section describes a new approach based on data mining, which is applicable for detection of gradual deterioration of behaviour in discrete manufacturing pieces of equipment dedicated to transportation of parts between the processing workstations. Monitored energy patterns are correlated with workload data (number of pallets occupying conveyor segments at a time). Detection of gradual conveyor misalignment relies on checking whether the output of a rule based engine mapping input workload data into classes match with the output of a Support Vector Machine classifier operating on the energy values. The goal of the work described is to characterize the behaviour of testbed pieces of equipment from the viewpoint of energy consumption, and use the defined patterns in order to compare designer expectations with data values coming in real time from the line.

4.1.1. Experiment of correlation between power consumption values and workload

The data relevant for our discussion is related to energy consumption of the piece of equipment, and respectively its workload. The conveyor belts are running continuously, irrespective of whether the pallets residing on them are stopped via stoppers or not. When stoppers are in use to let NFC readers capture data of NFC tags on each pallet, there is an increase in friction between the conveyor belt and the pallet. Since the pallet is blocked from moving while conveyor is turning under the pallet, consequently there is more loads on the conveyor engine. It results in an observed increase of power consumption in the conveyor engine. Figure 19 depicts the power consumption of the engine rotating the bypass conveyor of FASTory Cell 5. Increases in engine power consumption are generally correlated with increases in the number of pallets on the conveyor belt. According to Figure 19 data samples with blue colour point to power consumption values of cell 5 conveyor system while the number of pallets on the conveyor belt is increasing from zero to five. It is obvious that increase in the number of pallets indicates increase the power consumption values. Data represented in red colour, demonstrate the engine power consumption while the belt is tightened due to misalignment. Such observed difference between normal condition and tightened status of conveyor belt, established the core of presented approach in this work.

It is worth mentioning that normal data sampled by E10 Energy Analyzer expansion module embedded into S1000 controller; do not show such a significant difference (stepwise) between power consumption values linked with various number of pallets occupying the conveyor. In order to make the shift over power data values more transparent, data have been expanded and amplified by winding the current wire 3 times more around the CT.

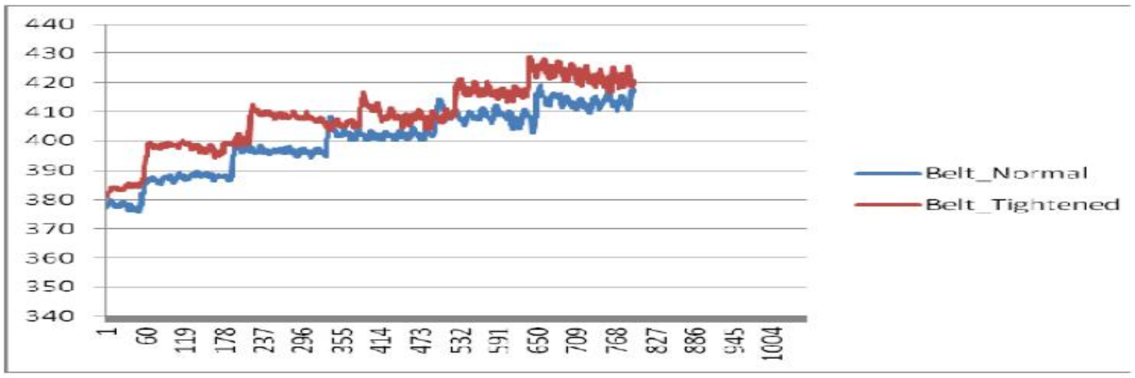


***Figure 19**. Cell 5 bypass conveyor engine power consumption. Pallet traffic of 0 to 5. In red: data obtained with the belt tightened*

4.1.2. Proposed approach for fault detection

Figure 20 illustrates the method employed for detection of gradual undesired behavioural changes in the considered equipment piece. The main factors involved are a monitor, to collect raw values from the line, a classifier for energy data and a rule based engine defined offline: The monitored data (energy values e(k) coming from the conveyor, and respectively the device workload NoPallets(k) in number of pallets occupying the conveyor at one moment of time t(k)) is collected from the testbed workcell. The energy values are input to an LS-SVM classifier (2.5.2) who categorizes the one dimensional data into two classes Cm, m is either 1 or 2, corresponding to energy values correlated to 0-1and respectively 2-3 pallet workloads. 70% of data are used as a training set (tuning the classifier by finding suitable the parameters) and the remained 30 % are utilized for cross validation. The training set providers the SVM classifier with the model parameters. The model is then verified via the remaining 30% of the data. The Rule-based Engine includes two simple rules defined offline. The rules associate the monitored number of pallets to either class of type 1 (for 0 to 1 pallets detected) or class of type 2 (for 2 to 3 pallets). The results coming from the classifier are compared to the results coming from the rule based engine at each time instant considered (m is compared with n). Gradual increase in the number of consecutive mismatches between the two outputs would imply gradual deterioration of expected behaviour in the monitored piece of equipment. The energy values observed no longer correlate an expected to the semantics defined statically.

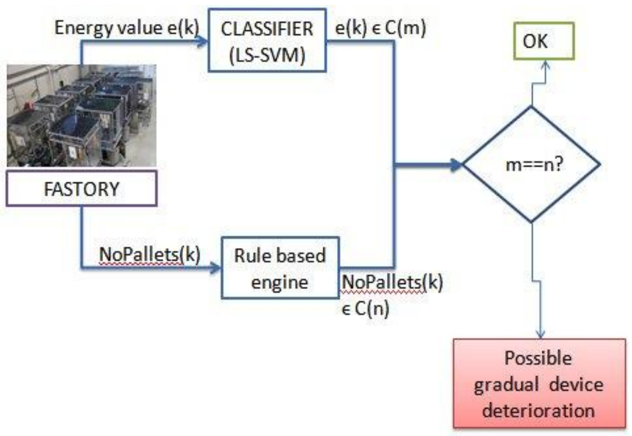


Figure 20. *Energy awareness for detection of gradual conveyor missalignment*

4.1.3. Empirical results of the proposed approach

A classifier was built to generate a model estimating the class of new energy values based on previous observations. The power consumption (WATT) of the conveyor system was measured at a sampling rate of 1 second. Figure 21 shows the data monitored (2500 power consumption data samples) during the operation of the conveyor system (including main and bypass conveyors). The most significant power consumption change is observed when the number of pallets changes from one to two on the conveyor system. Consequently, the first class corresponds to power consumption values estimated to be representative of zero or one pallet and the second class corresponds to power consumption values representative for two, three or more number of pallets.

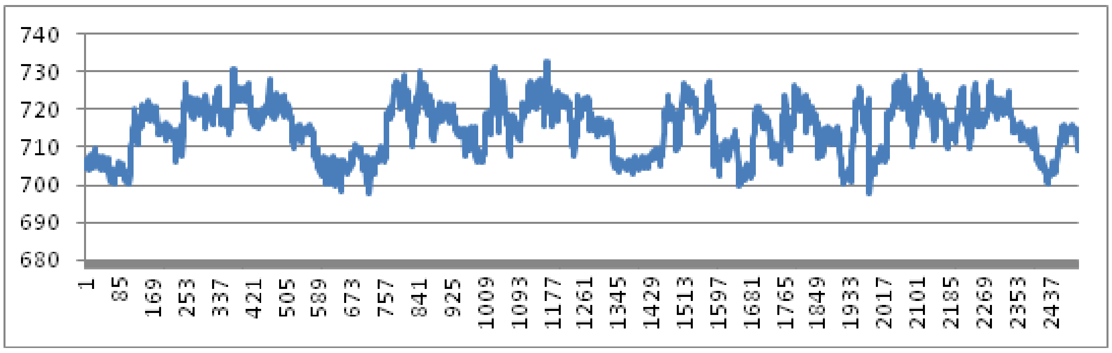


Figure 21. *Cell 5 conveyor system engine (main and bypass conveyor) power consumption*

Figure 22 illustrates the two classes identified by the rule based engine on the sampled data represented in Figure 21 . The data categories are separated based on observed power consumption values and number of pallets associated to each sampled power value.

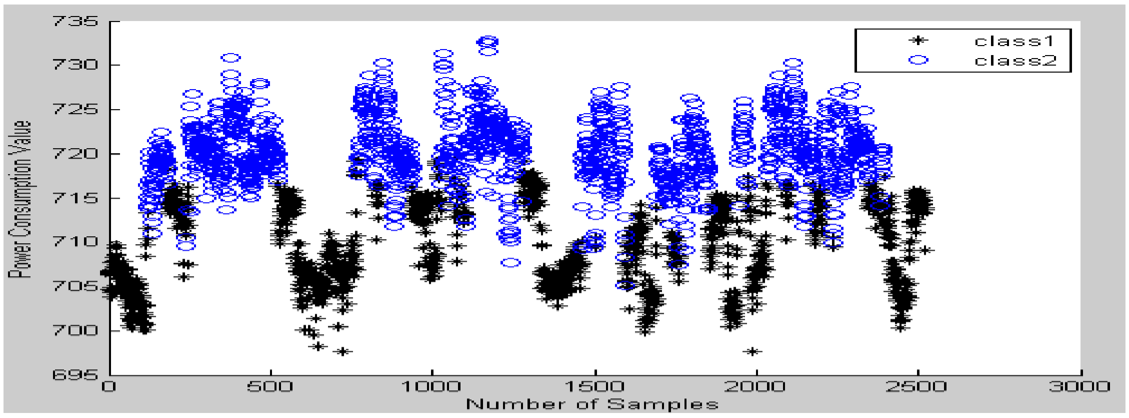


Figure 22. *Classes generated by rule based engine and correlated to each sampled data*

A binary classifier algorithm (LS-SVM) with radial basis kernel function (2) was applied to the data shown in *Figure 22* .

$$K(x,z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right) \qquad (2)$$

1800 data samples were used for training, and 700 for validation. The accuracy of the classifier performance is evaluated by computing the error term defined as the fraction of the cross validation examples that were classified incorrectly. In our experiment, the computed error is 5.56%. This error is achieved by calculating the average percent of the number of unsuccessful estimated classes compared with the classes given by rule based engine. *Figure 23* shows the cross validation data classified into two classes by LS-SVM.

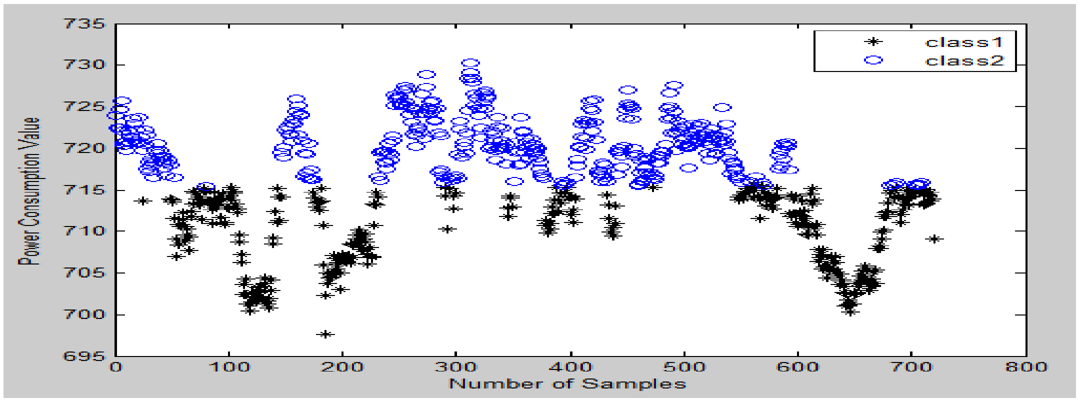


Figure 23*. Classified cross validation data generated by LS_SVM*

Chain consecutive mismatches of new data against classes generated by rule based engine should pinpoint to incipient gradual deterioration of expected behaviour. The method applied for calculating the error for classifier performance through cross validation data is applied for fault detection as well. In the presented scenario, such deterioration would translate to a misalignment of conveyor segments.

An important issue of this approach is to avoid the generation of an illusion of causality by an improperly tuned classifier. That is, an inappropriately large classifier error would immediately result in mismatches between the output of the rule based engine and the output of the LS-SVM, mismatches that would not be caused by conveyor deterioration but by the classifier's inability to associate correct categories to the data it is presented with.

Despite expectations, conveyor power consumption does not change instantly once conveyor workload is modified, but after a short time delay. This delay is responsible for the few outliers visible in *Figure 22*. For instance, the number of pallets is 2, one pallet leaves the cell, then the counter shows number 1, but it takes some time for conveyor engine to show a power consumption value matched with the new number of pallets. Such delays on power consumption changes may influence the value of the calculated error. One of the solutions to reduce the number of outliers is to increase the sampling time. It provides more time delay and increases the probability to sample power consumption value after modification toward the workload alterations. However, increasing the sampling time leads to lose more data.

The Future research will focus on bringing more parameters for analysis, in addition to power consumption, to increase the number of dimensions of the available datasets. Vibration and temperature sensors are available in the testbed, and can be used wherever applicable (e.g. for the robots). As SVMs successfully support regression and classification of high dimensional input spaces, they were used here to provide an implementation backbone for future work. Moreover, the classifier performance is more accurate when dealing with higher dimension of data sets.

4.2. Application of process mining in factory automation

Data/question and goal driven process mining techniques are applied on the event logs produced by the FASTORY line. This work focuses on process mining covering 3 out of 8 ISA-95 level 3/MES (production/manufacturing) activities (*Figure 24*):

- *Production tracking*: Following the movement of materials (pallets) in factory line using events translated from CAMX notification messages. This activity is entirely covered by work-flow (process) perspective of the process mining. This activity also provides production performance reports for business level (level 4). It is also linked with *detailed production scheduling* task. The discovered and real behaviour of the processes is rendered to *detailed production scheduling* activity in order to evaluate the planned production and perform necessary modifications.
- *Production performance analysis*: engineering functions through production data such as evaluating the performance of different machines in order to assess the balance of the line and discover bottlenecks. This activity also focuses on the

‘golden runs’ , discovering the exceptional conditions in which excellent results achieved, to reproduce the same conditions. Performance analysis is one of the typical capabilities of process mining.

- *Production data collection*: the most important activity of manufacturing and operation level. *Production data collection system* is responsible for collecting, storing and reporting information about the execution of processes in factory floor. In our case, data collection is performed in conjunction with Complex Event Processing in order to store the data in desired format including required attributes. Historical data are stored in a remote MySQL data base comprising raw events.

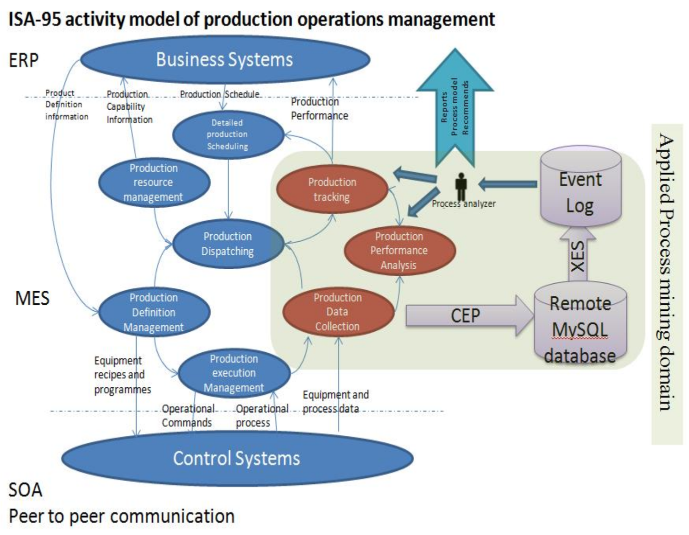
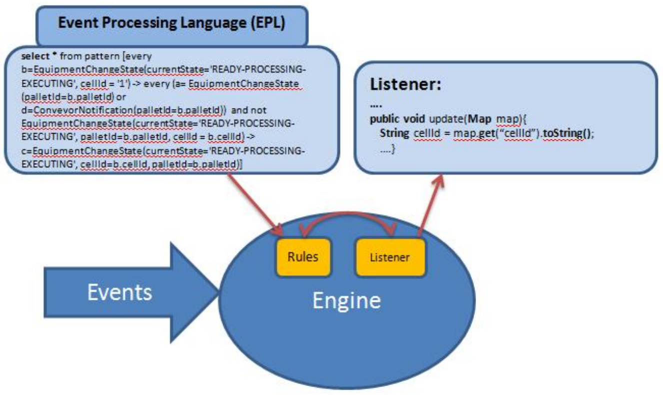

ISA-95 activity model of production operations management
ERP
Business Systems
Product Definition information
Production Capability Information
Production Schedule
Production Performance
Reports
Process model
Recommends
Detailed production Scheduling
Production resource management
Production tracking
Process analyzer
Event Log
Applied Process mining domain
Production Dispatching
Production Performance Analysis
XES
MES
Production Definition Management
Production Data Collection
CEP
Remote MySQL database
Equipment recipes and programmes
Production execution Management
Equipment and process data
Operational Commands
Operational process
Control Systems
SOA
Peer to peer communication

Figure 24. *ISA-95 architecture focusing on level 3*

4.2.1. Data collection and event log preparation (inspection/cleaning)

Figure 25 depicts the architecture of the monitoring application concerned with event processing and storage.



```
select * from pattern [every
b=EquipmentChangeState(currentState='READY-PROCESSING-
EXECUTING', cellId = '1') -> every (a= EquipmentChangeState
(palletId=b.palletId) or
d=ConveyorNotification(palletId=b.palletId)) and not
EquipmentChangeState(currentState='READY-PROCESSING-
EXECUTING', palletId=b.palletId, cellId = b.cellId) ->
c=EquipmentChangeState(currentState='READY-PROCESSING-
EXECUTING', cellId=b.cellId, palletId=b.palletId)]
```

```
....
public void update(Map map){
  String cellId = map.get("cellId").toString();
   ....}
```

Figure 25. *Applied method for storing the data including necessary parameters*

The Engine is a JAVA class defining the Event processing Language rules responsible for receiving the real time data of interest from FASTORY.

Table 7

Defined rules of event processing

Rule	Content of the rule	Description
b	b=EquipmentChangeState(currentState='READY-PROCESSING-EXECUTING', cellId = '1')	Robot notification that the IPC2541 state of Cell 1 has now been found to be'READY-PROCESSING-EXECUTING' . This is the first notification message coming from Cell 1.
a	a=EquipmentChangeState(palletId=b.palletId)	Robot (rule a) and conveyor (rule d) notification messages which contain the palletID similar with rule b .
d	d=ConveyorNotification(palletId=b.palletId)	
c	c=EquipmentChangeState(currentState='READY-PROCESSING-EXECUTING', cellId=b.cellId, palletId=b.palletId)	Notification that the IPC2541 state of robot has found, if cellID and palletID numbers are the same as cell 1 notification message which had satisfied rule b.

Since productID is an important parameter in order to distinguish between messages related to different pallets and to generate the traces/processInstances, complex event processing applied at data capturing level. The first cell of FASTORY line sends the first notification message declaring the state of the robot (READY-PROCESSING-EXECUTING). This message satisfies rule b. After that by moving the pallet among the cells, notification messages of robot and conveyors are sent. A distinct productID number is assigned to all the messages fulfilling rules a and d. This procedure continues until pallet arrives to the cell 1 which means that product is ready to be unloaded. At this stage rule c is satisfied because the palletID is similar to the one fulfilled rule b at the beginning. If there are multiple pallets at the same time moving on the line, each one includes a distinct productID provided when rule b was satisfied. In conclusion, all messages complying with rules a or d, and captured between notification outputs of rules c and b, specified by a productID and stored in the Listener, and subsequently in a MySQL database.

Table 8 illustrates the format of the raw data collected from the FASTORY line, for one process instance. Each product is assigned an orderID (in the presented case the value of this ID is 1), corresponding to the XES trace and respectively the MXML processInstance, and a sequence of visited cellIDs.

Table 8

Raw data associated with producing one product stored in MySQL database

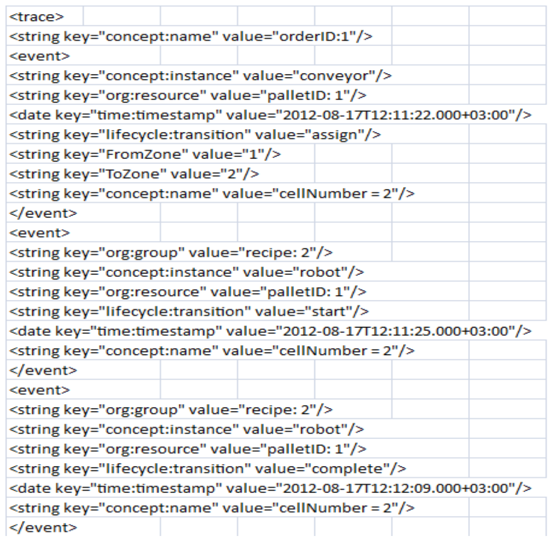
orderID	cellID	Pallet-State	timeStamp	recipe	palletID	fromZone	toZone	device
1	2	assign	17/08/2012 12:11:22	0	1	1	2	conveyor
1	2	start	17/08/2012 12:11:25	2	1			robot
1	2	complete	17/08/2012 12:12:09	2	1			robot
1	3	autoskip	17/08/2012 12:12:16	0	1	1	4	conveyor
1	4	autoskip	17/08/2012 12:12:26	0	1	1	4	conveyor
1	5	assign	17/08/2012 12:12:42	0	1	1	2	conveyor
1	5	start	17/08/2012 12:12:48	4	1			robot
1	5	complete	17/08/2012 12:13:32	4	1			robot
1	6	autoskip	17/08/2012 12:13:38	0	1	1	4	conveyor
1	8	assign	17/08/2012 12:13:54	0	1	1	2	conveyor
1	8	start	17/08/2012 12:13:57	7	1			robot
1	8	complete	17/08/2012 12:14:18	7	1			robot
1	9	autoskip	17/08/2012 12:14:27	0	1	1	4	conveyor
1	10	autoskip	17/08/2012 12:14:37	0	1	1	4	conveyor
1	11	autoskip	17/08/2012 12:14:47	0	1	1	4	conveyor
1	12	autoskip	17/08/2012 12:14:58	0	1	1	4	conveyor

The XESame tool [100], [18] was used to generate XES format event logs from such raw data. *Figure 26* illustrates partly the resulting event log obtained. A number of extensions (Table 9) were applied in the process.

Table 9

Applied XES Standard extensions

Type of extension	Key	Value	Originator	Description
Lifecycle	Transition	*assign*	Conveyor message	Pallet is on the entrance stopper of the cell, it will go to the main conveyor of the cell for operation. from zone 1 to zone 2
Lifecycle	Transition	*autoskip*	Conveyor message	Pallet is on the entrance stopper of the cell, it will go to the bypass conveyor of the cell for .zone 1 to zone 4
Lifecycle	Transition	*start*	Robot message	Pallet under operation by robot
Lifecycle	Transition	*complete*	Robot message	Pallet'soperation is completed
Orgenizational	group	recipe	Robot&conveyor	The recipe that robot may operate on the pallet
Orgenizational	resource	palletID	Robot&conveyor	Each pallet has an ID number used for producing the orderIDs
Concept	Name	cellID	Robot&conveyor	Stores a generally understood name for any type hierarchy element; trace,log,event
Concept	instance	device	Robot&conveyor	It represents an identifier of the activity instance whose execution has generated the event.



```
<trace>
<string key="concept:name" value="orderID:1"/>
<event>
<string key="concept:instance" value="conveyor"/>
<string key="org:resource" value="palletID: 1"/>
<date key="time:timestamp" value="2012-08-17T12:11:22.000+03:00"/>
<string key="lifecycle:transition" value="assign"/>
<string key="FromZone" value="1"/>
<string key="ToZone" value="2"/>
<string key="concept:name" value="cellNumber = 2"/>
</event>
<event>
<string key="org:group" value="recipe: 2"/>
<string key="concept:instance" value="robot"/>
<string key="org:resource" value="palletID: 1"/>
<string key="lifecycle:transition" value="start"/>
<date key="time:timestamp" value="2012-08-17T12:11:25.000+03:00"/>
<string key="concept:name" value="cellNumber = 2"/>
</event>
<event>
<string key="org:group" value="recipe: 2"/>
<string key="concept:instance" value="robot"/>
<string key="org:resource" value="palletID: 1"/>
<string key="lifecycle:transition" value="complete"/>
<date key="time:timestamp" value="2012-08-17T12:12:09.000+03:00"/>
<string key="concept:name" value="cellNumber = 2"/>
</event>
```

Figure 26*. Fragment of generated event log by XESame tool*

Real life process event logs usually include noise, or exhibit lack of necessary data. *Log inspection* refers to examining the log entry types to determine whether all data necessary is included and whether there are noisy pieces of information (i.e. outliers data representing uncommon behaviours in the event log), that are not needed for later analysis. Analysis of the Fastory event log via ProM's Inspector tab showed 147 cases (traces/processInstances), 2563 events , 6 originators (i.e. pallets), 4 event types (*Start*, *Complete*, *Autoskip* and *Assign*), 40 classes (10 cells x 4 event types)

Log cleaning aims at generating a new event log based on the original one, where log inspection conclusions are taken into account for determining final data entries. ProM's log filter plug-in was utilized on the event log of FASTory to artificially add START and END events at the beginning and end of each trace. This is needed because in the testbed considered, cell 1 serves both as entry and exit point to/from the manufacturing process. It is not possible to conclude that a pallet is at the beginning or at the end of the production process, if this conclusion relies solely on the raw monitored data. At log cleaning stage, an orderID parameter is introduced and mapped to each trace.

4.2.2. Process Mining for Discovery: Results

A number of ProM's discovery plug-ins (e.g. the alpha miner, the heuristic miner, the genetic miner, the fuzzy miner, the transition system miner, etc.) are applicable on the input event logs from FASTory.

Examples of questions that may be answered based on the input event log include:

- How is the actual execution of the cases?
- How many components exist in each case?
- What kind of patterns are in the log?
- How is the dependency between different activities?
- How different sections of the process are communicating?
- How many executions happened between different activities?

- What are the similarities between cases or which parts are working similar to each other?
- If some rules exist between events or activities, are they satisfied?

4.2.2.1 Discovery based on the alpha algorithm

ProM's *Alpha Miner* plug-in produces a Petri Net (PN) model based on a given event log. The control flow model obtained from the event log of FASTory (Figure 27) shows the event notifications of the cells as transitions. Events belonging to the same cell are grouped via blue rectangles.

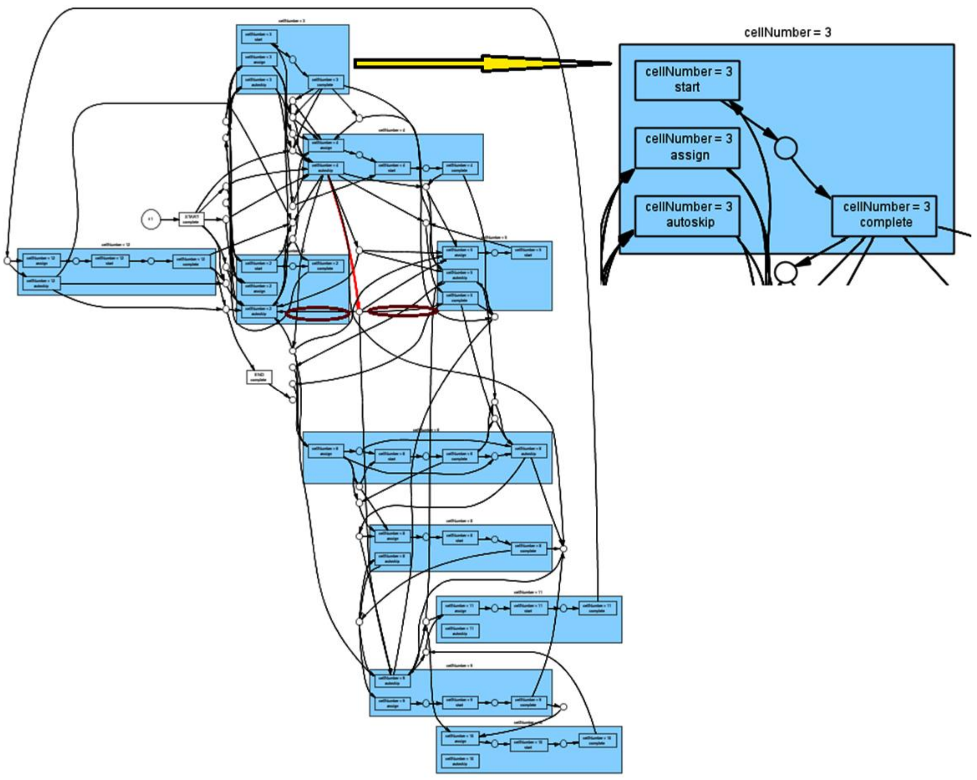


Figure 27. *Alpha algorithm applied to Fastory event log: output PN model. Cell 3 (right side) exhibits 4 types of messages; autoskip, assign, start and complete messages.*

The PN model generated via alpha algorithm exhibits one unexpected transition (highlighted in red, in the figure). This transition shows possibility for the pallet to move back to Cell 2, after bypassing from Cell 4, or to directly go to the *complete* state of Cell 5. This is in disagreement to the real life situation (the actual control embedded in the line), depicted in *Figure 28*. Pallet flow in Fastory is unidirectional (Cell1 → Cell2 → Cell3 → Cell4 → Cell5 →Cell1). Pallets at the entry point of a cell can either go to the main conveyor for processing by the robot (corresponding to a sequence of messages *assign*, *start*, *complete*), or to the bypass conveyor of *that* cell (i.e. one *autoskip* message).

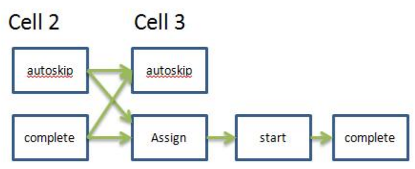


Figure 28. *Feasible message sequence patterns between consecutive cells in Fastory*

ProM’s *Conformance Checker* plug-ins built to evaluate the fitness of the generated model and its conformance with the event log. Three different types of analysis are embedded: fitness, precision and structure.

Fitness is calculated by replaying the log on the Petri net format of the model. It evaluates *whether* the observed process complies with the control flow specified by the process.

Precision/Behavioural Appropriateness evaluates *how precisely* the model describes the observed process. ´

Structural fitness evaluates whether the model describes the process in a structurally suitable way.

The fitness of the model shown in *Figure 27* was evaluated at approx. 58%. This result is in agreement to findings reporting unreliability exhibited by the α-algorithm in real life projects[29], [53]. The *Alpha miner* is not suitable for FASTory’s event log.

4.2.2.2 **Discovery based on the heuristic approach**

The high fitness and precision of the models generated based on the heuristic approach, have given rise to tremendous interest by analysts to apply this algorithm. The strongest point of *Heuristic Miner* (2.6.1.2) is its robustness to noise.

The Heuristic Miner operates based on causal dependency between events.

Figure 29 shows the control flow model produced by *Heuristic miner* for the event log of FASTory. All traces start and end by the START/END activities added manually at log cleaning stage.

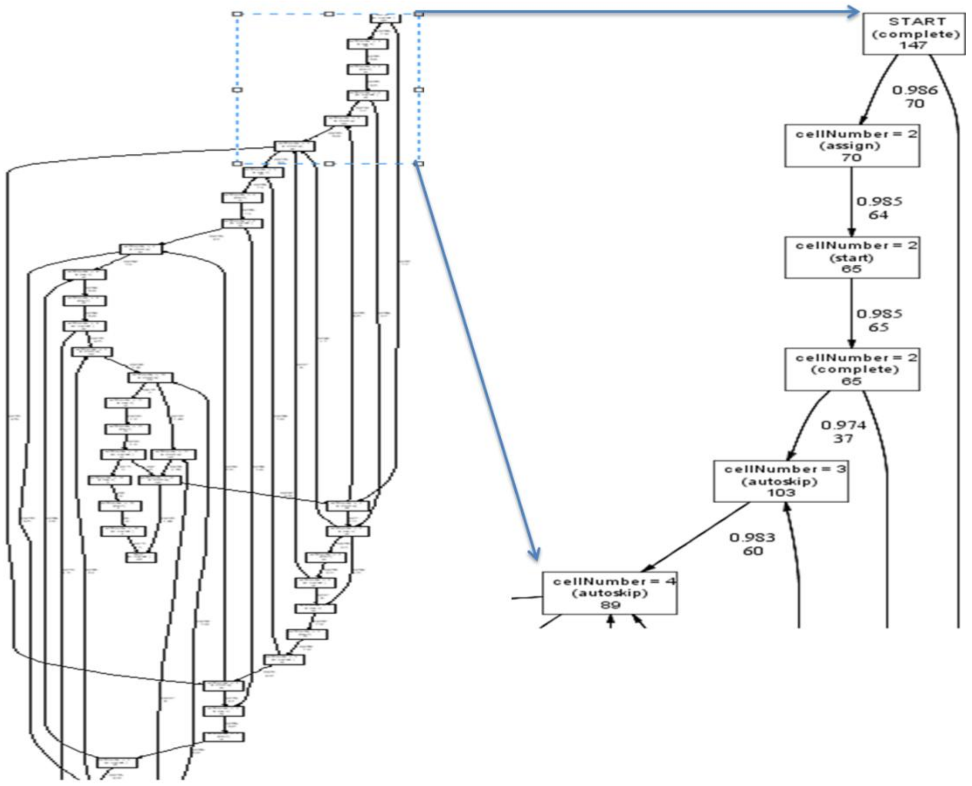


Figure 29. *ProM's Heuristic Miner applied to the event log of Fastory (Partial result). Zoom in on the right side.*

Tasks correspond to the boxes of the model, with connecting arcs showing dependency between the tasks. Numbers associated to each box indicate the amount of times the associated tasks have been performed (e.g. according to the model shown, the robot of Cell 2 has been assigned to operate on the pallet 70 times). Numbers associated to the arcs indicate arc frequency (the number of times the tasks associated to the arc performed sequentially; bottom) and the dependency (i.e. how certain we are that two activities are dependent of each other; top).

To evaluate model fitness and conformance via the *Conformance Checker*, the input must be formerly converted to a PN (*Figure 30*), or EPC / Fuzzy models (refer to).

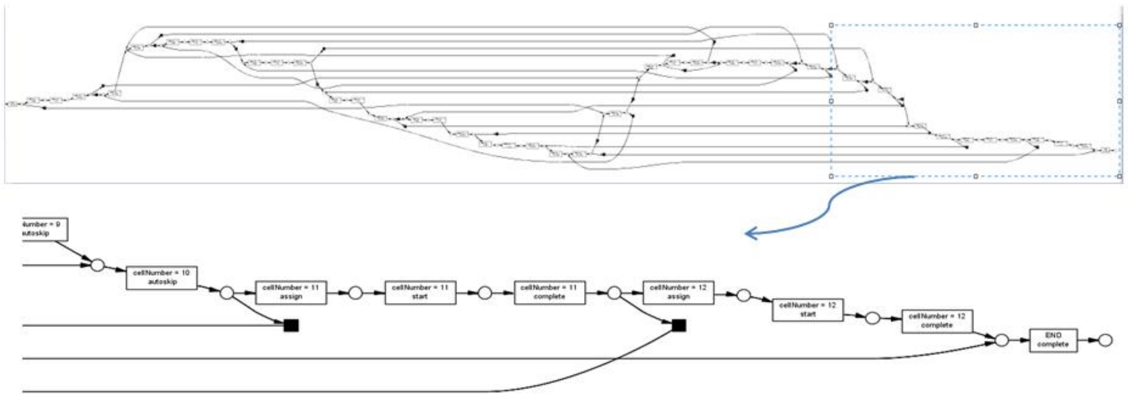


Figure 30. *The Heuristic model of Figure 24, converted to Petri Nets*

The fitness value computed for the obtained heuristic net model is 98%, confirming the model reflects accurately the real testbed scenarios.

4.2.3. Basic Performance Analysis

ProM's *Basic Performance Analysis* plug-in calculates performance measures such as execution (working) time, waiting time and graphically displays the results obtained. Such graphs are helpful for evaluating the effects of the control scenario on the workload balance of the line.

Figure 32 shows the total processing time of each robot via bar charts. X axis represents the cell number and Y axis represents total working time of robots for each cell in seconds. According to *Figure 32*, for 146 products, there is no balance in the line. Figure 31 is another representation of results in pie chart .*Figure 31* is illustrating that for 146 produced items, robots of cells 2,4,5,6 have outperformed other robots (unsteadiness of the production load over the cells). This information may be input to support scheduling of robot maintenance time under the same control scenario.

4.2.3.1 General performance of the line

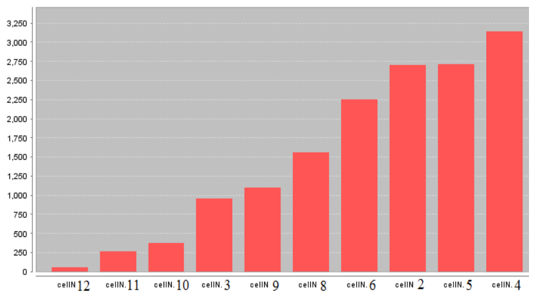


***Figure 32**. Basic performance analysis (x axis - performers; y axis – total cell working time, seconds)*

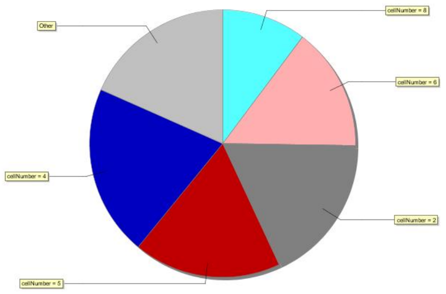


***Figure 31**. Pie chart, basic performance analysis*

Several KPIs designed and implemented at FASTORY line in order to supervise the success of goals. Those KPIs are under measurement frequently to provide proper knowledge for fast decisions. To handle and overcome high quantity of real time data coming from lower levels, proper and powerful information technology must be applied at higher levels. Service Oriented Architecture (SOA) and Complex Event Processing (CEP) are two IT systems applied at higher level. SOA provides fast and reliable capability to collect information holding status of manufacturing processes and CEP is applied for tracking and analyzing stream of that information in order to store events in a desired format in databases. Implemented KPIs at FASTORY are classified into efficiency, Energy, Quality and Reliability and some other KPIs reflecting the overall status of the entire production line.

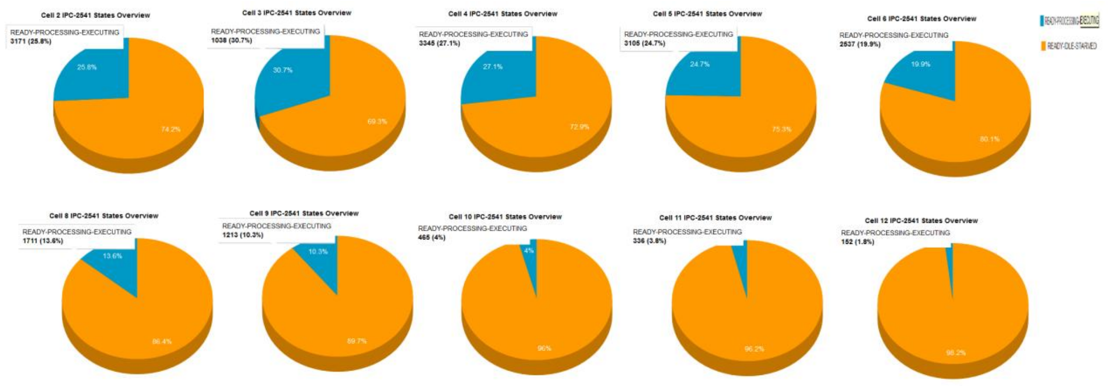


Figure 33. *KPIs including IPC-2541 states overview*

The IPC-2541 states overview is rendered as a 3D pie chart (*Figure 33*) with each portion of the pie representing the duration of a state. KPIs shown in *Figure 33*are representing only READY-PROCESSING-EXECUTING (robot is operating) and READY-IDLE-STARVED (robot is idle) states of the robots. As soon as one state completes, the corresponding portion increases accordingly.

Table 10

Comparison of online KPIs with KPIs applied offline for basic performance analysis

Robot	IPC-2541 states overview Robot working time(sec)	Process mining Basic performance analysis Robot working time(sec)
2	3171	2708
3	1038	960
4	3345	3146
5	3105	2724
6	2537	2263
8	1711	1566
9	1213	1105
10	465	384
11	336	275
12	152	66

Table 10 illustrates the results of online KPIs (pie charts) showing the IPC-2541 status of the robots and KPIs applied offline for performance analysis of the FASTORY. Comparing the results of them demonstrates that they have similar results and robots 2,4,5 and 6 are surpassing other robots. Given the results from both types of KPIs, some trivial dissimilarity in the working time of robots are visible in both columns. Those differences are due to the fact that the rules applied in the CEP of online and offline

KPIs are slightly different. As explained in section 0, in order to provide event logs for process mining techniques, events are stored in the listener when they satisfy rules c and d, otherwise all the messages related to the related productID are not recorded. In other words, notification message of each productID will be stored in a chain if pallet starts from cell number one and ends to cell number one. In contrast, datasets applied for online KPIs are including whole data because there is no need for productID parameter (online KPIs have not developed for process analysis point of view). Consequently, although online KPIs are representing that working time of robots is slightly higher than offline KPIs, they demonstrate similar results about performance and load balance of the FASTory.

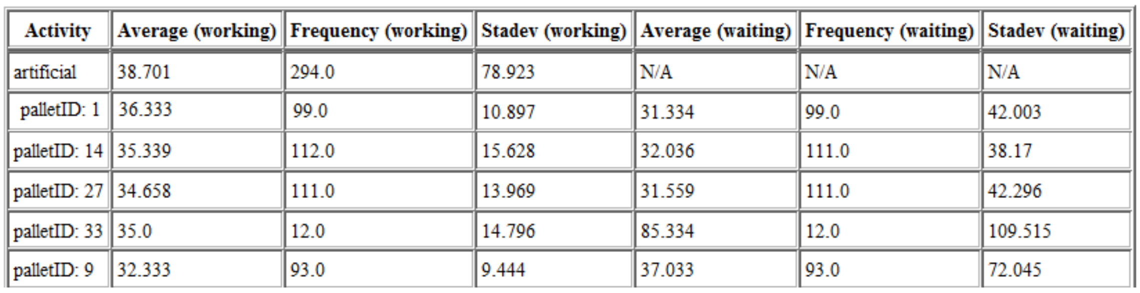
4.2.3.2 Performance analysis of the pallets

After evaluating the performance of the robots and balance of the line (4.2.3.1), pallet performance is also another option. In case of mass production, a large number of pallets are applied and tracking everything online is both time consuming and expensive. Usually it is not possible to detect the pallets malfunctioning (i.e. uneven movement / off-balance structure of pallets) by monitoring due to traffic of the pallets or some delay problems. Malfunctioning of the pallets leads to various problems such as decrease in line efficiency, increased traffic /bottlenecks, damaging of other pallets, increased maintenance costs.

An analysis example is illustrated in *Table 11*. The interval for a pallet moving among two sequential FASTory cells is usually less than 20 seconds. The longest path between two sequential cells (25 seconds) is between cells number 6 and 8, because cell number 7 of the FASTory line is under development and bypassed by a conveyor.

Table 11

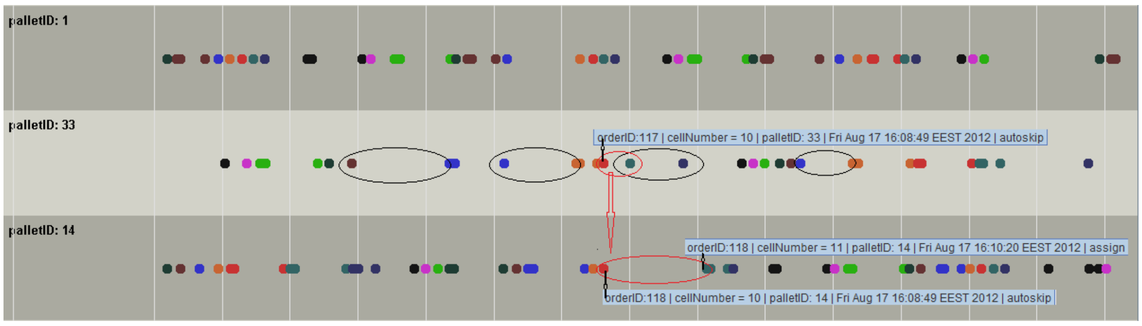
Pallet performance (Basic Pefrormance Analysis plugin, text view)



Activity	Average (working)	Frequency (working)	Stadev (working)	Average (waiting)	Frequency (waiting)	Stadev (waiting)
artificial	38.701	294.0	78.923	N/A	N/A	N/A
palletID: 1	36.333	99.0	10.897	31.334	99.0	42.003
palletID: 14	35.339	112.0	15.628	32.036	111.0	38.17
palletID: 27	34.658	111.0	13.969	31.559	111.0	42.296
palletID: 33	35.0	12.0	14.796	85.334	12.0	109.515
palletID: 9	32.333	93.0	9.444	37.033	93.0	72.045

Inspection of *Table 11* leads to the conclusion that the parameters *Average(working)* and *Frequency* are contradicting each other for Pallet 33: while the frequency of execution is lower for Pallet 33 compared to other pallets, its average working/waiting time for is high.

Figure 34 shows the dotted chart view of the log. Each row of data stands for each pallet number, and each dot of a specific colour represents one cell. The dots circled by oval shapes correspond to the moments that pallets had significant delay while moving between the two cells. It is visible that the time interval for pallet 33 while moving between the cells (on intermediate bypass conveyors) exceeds the expected time in several instants. Improper operation of some pallets affects other pallets' performance. The red arrow (*Figure 34*) highlights the improper operation of pallet 14 (a significant delay, of approx 1 min 30 seconds, until the following message is received from Pallet 14). Cell number 10 (red dot) contains both pallets 33 and 14 at the same time on the bypass conveyor (*autoskip*). It can be concluded that pallet 33 is influencing other Pallet 14's performance.



orderID:117 | cellNumber = 10 | palletID: 33 | Fri Aug 17 16:08:49 EEST 2012 | autoskip

orderID:118 | cellNumber = 11 | palletID: 14 | Fri Aug 17 16:10:20 EEST 2012 | assign

orderID:118 | cellNumber = 10 | palletID: 14 | Fri Aug 17 16:08:49 EEST 2012 | autoskip

***Figure 34**. Pallets performance. Dotted chart analysis*

Figure 35 shows some other effects of Pallet 33 having inappropriate operation. It is visible that pallet 33 has made traffic with pallets 1 and 9 in one cell (the orange colour dot).

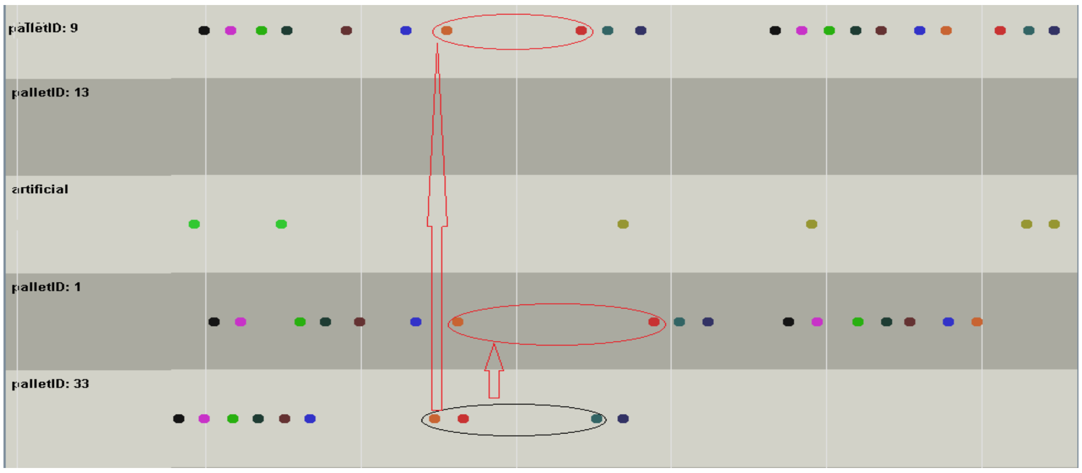


***Figure 35**. The effect of improper operation of pallet 33 on other pallets*

After highlighting the possibility of defect in pallet 33 using process mining, a subsequent inspection of the pallet pointed out that one of the bearings of the pallet was out of lubrication, leading to uneven movement.

4.2.4. Process Mining for Conformance: Results

4.2.4.1 Enhancement of the discovered model

This section documents results obtained when checking a reference process model against the event log of the line.

Figure 36(a) illustrates the model of *Figure 30*, after the log entries are replayed on it in ProM's *Conformance Checker*. The information obtained includes the number of missing / remained tokens, the tasks that could not be enabled in replay mode, the tasks enabled but unfinished, and path coverage (i.e. all tasks and arcs involved in replaying the event log).

Some of the questions of interest here include:

- Does this model properly represent the actual process?
- How much of the observed model behaves like the reference model?
- If some deviations exist between the observed model and the reference one, where are they located in the model?

The orange highlighted boxes of Figure 30.a are illustrating those parts of the model that do not comply with the log or the reference model. The red highlighted circles show the remaining / missing tokens when replaying of the log.

Given *Figure 36* (b), the textbox associated to cellNumber=9 (autoskip) containing #token and # instances information , details the fact that for 3 out of 144 product instances (papers/ processInstances/traces), after the log is replayed on the reference model, there is an inconsistency found between the updated model and the event log (cell number 9 (autoskip) event is triggered in the event log despite impossibility to do so when inspecting the model – deficiency in input tokens associated to the transition mirroring the firing of the mentioned event). The number of missing tokens associated to the deficient input place(s) is specified in the #tokens column, meaning the transition is fired in the replay log mode despite the lack of tokens enabling it.

The log perspective is more suitable for highlighting discrepancies. *Figure 37* shows an example, depicting the sequence of the events in each trace (process instance) from beginning to end. OrderIDs associated to the traces are shown on the left side.

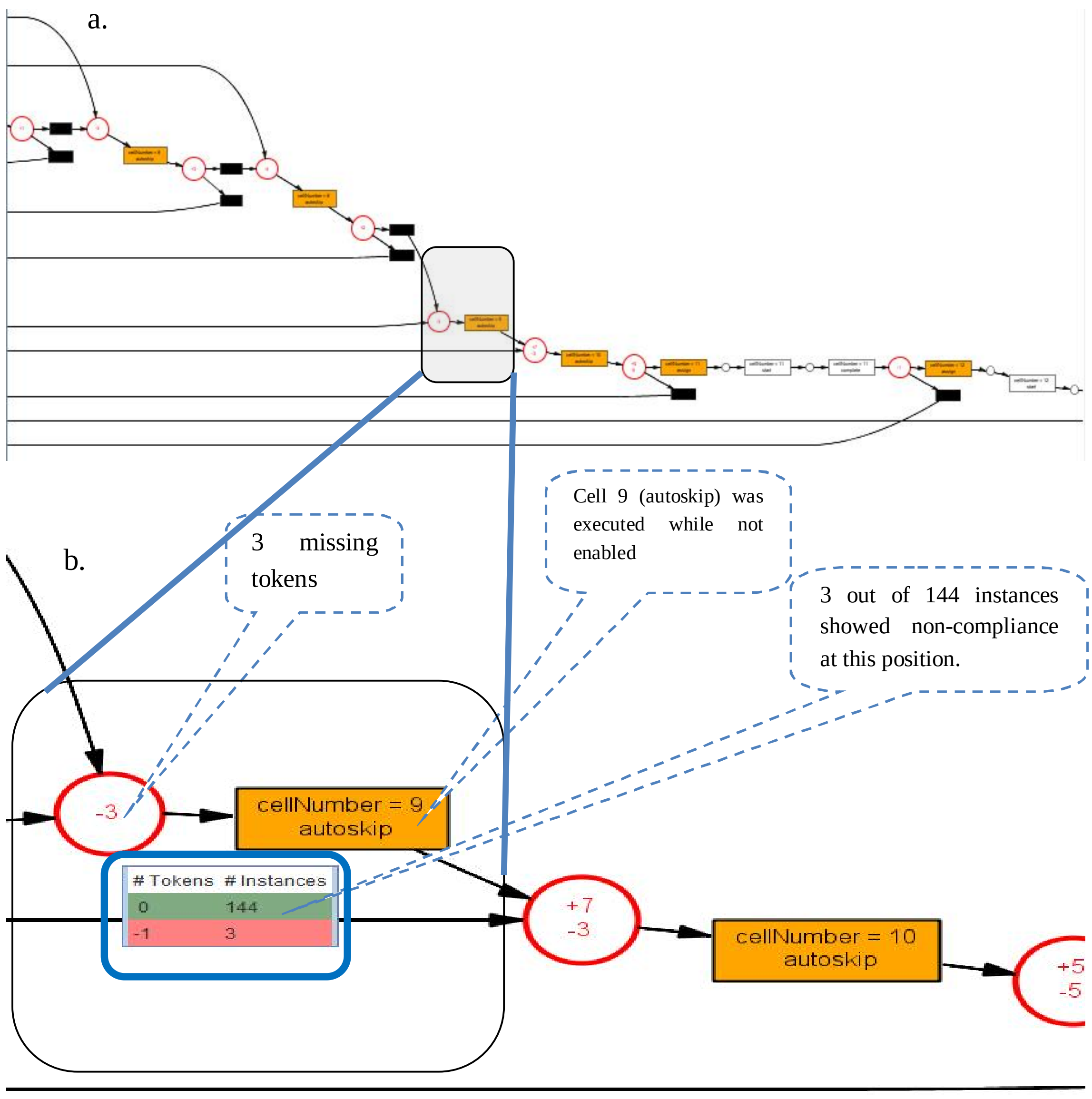

a.
b.
3 missing tokens
Cell 9 (autoskip) was executed while not enabled
3 out of 144 instances showed non-compliance at this position.
-3
cellNumber = 9 autoskip
# Tokens # Instances
0 144
-1 3
+7 -3
cellNumber = 10 autoskip
+5 -5

Figure 36. *a : The Conformance Checker measures and visualizes discrepancies between the log and a given process model. Model Perspective of the result obtained for replaying the event log entries on the PN of Figure 30. b: zoom view with descriptions*

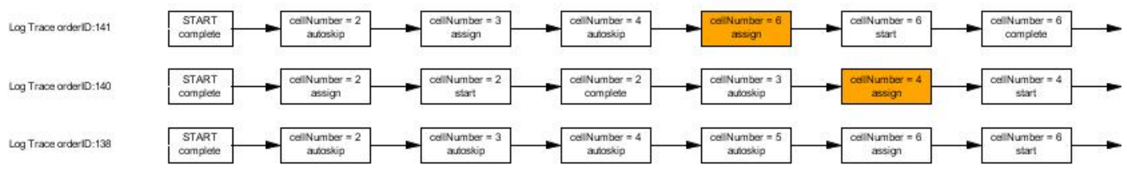


Figure 37 *. log diagnostic perspective*

The events unfeasible to be replayed on the generated model are highlighted in orange. Examining those highlighted events via both model and event log might lead to improvement of the model or detection of incomplete instances due to lost messages:

The orange event for OrderID 141 highlights that the messages of Cell 5 are missed, and therefore the *assign* event (move to main conveyor) of Cell 6 was not replayed on the model. Inspection of the contents of orderID 141 led to conclude no message from cell number 5 was even received for this trace/process instance.

The cause of the orange event for OrderID 140 is not immediately obvious (the sequence of messages complies with the rules designed for Fastory, therefore the event log is most probably correct). To investigate this discrepancy between the model and the real log, the transition from cellNumber3 (autoskip) to cellNumber4 (assign) is inspected in model perspective (*Figure 38*). Findings show the model includes only one transition from the *autoskip* event (bypass conveyor) of Cell 3 to the *autoskip* event of Cell 4. Therefore the model is not complete enough to replay all the events of the log.

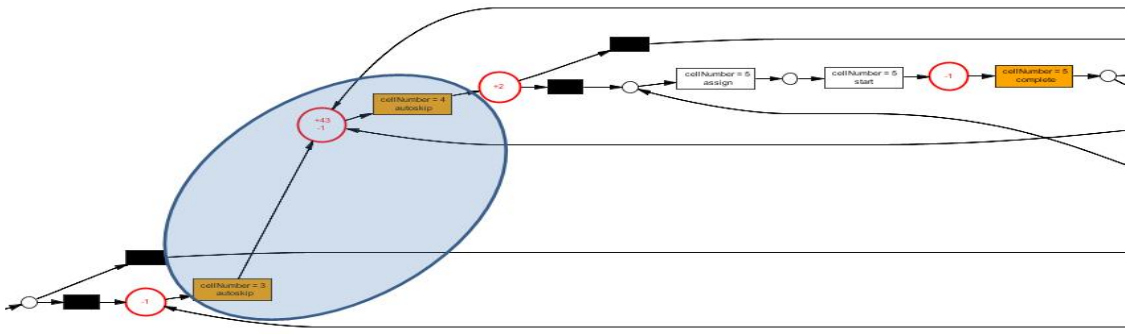


***Figure 38**. Noncompliance between the model and the log (highlighted in orange)*

Figure 39 shows that most non-compliance cases between the log and the model are associated with Cell 4. According to stage 3 of process mining structure (*Figure 6*) by applying conformance checking or comparing the predefined model with the discovered model, the model is verified or both models are combined to have a more precise reflection about the process activities. In our case, by adding a transition from cell cellNumber3 (autoskip) event to cellNumber4 (assign) event, it should be possible to replay all 'problematic' messages and it will result in a model which is representing the behaviour of the processes for producing 146 products, based on the event logs. This step is considered as the enhancement of the model.

Other possible sources of non-compliance between the model and the log are lost messages and/or communication problems leading to a wrong notification message format. Such causes do not arise frequently, and are considered a negligible source of errors for the test case of focus.

Note: Model *soundness*[81] , [30] , must be ensured prior to replaying the log on the model and checking conformance of the model against the event log. Three criteria are defined to evaluate soundness of a WF-net:

- *Complete option*: for each case, it is always possible to reach to a state defined as the END.
- *Proper completion*: When state is marked as END, none of the other places of the case is marked.
- *No dead transition*: there should not be any deadlock in the case, always there should be an auxiliary path in case a transition is impossible to be fired.

A WF-net is sound if and only if the short-circuited net (i.e. END and START places connected to each other) is both live and bounded [31]. Model soundness is ensured by artificially adding START/END events at the beginning/end of all traces via *Conformance Checker* .

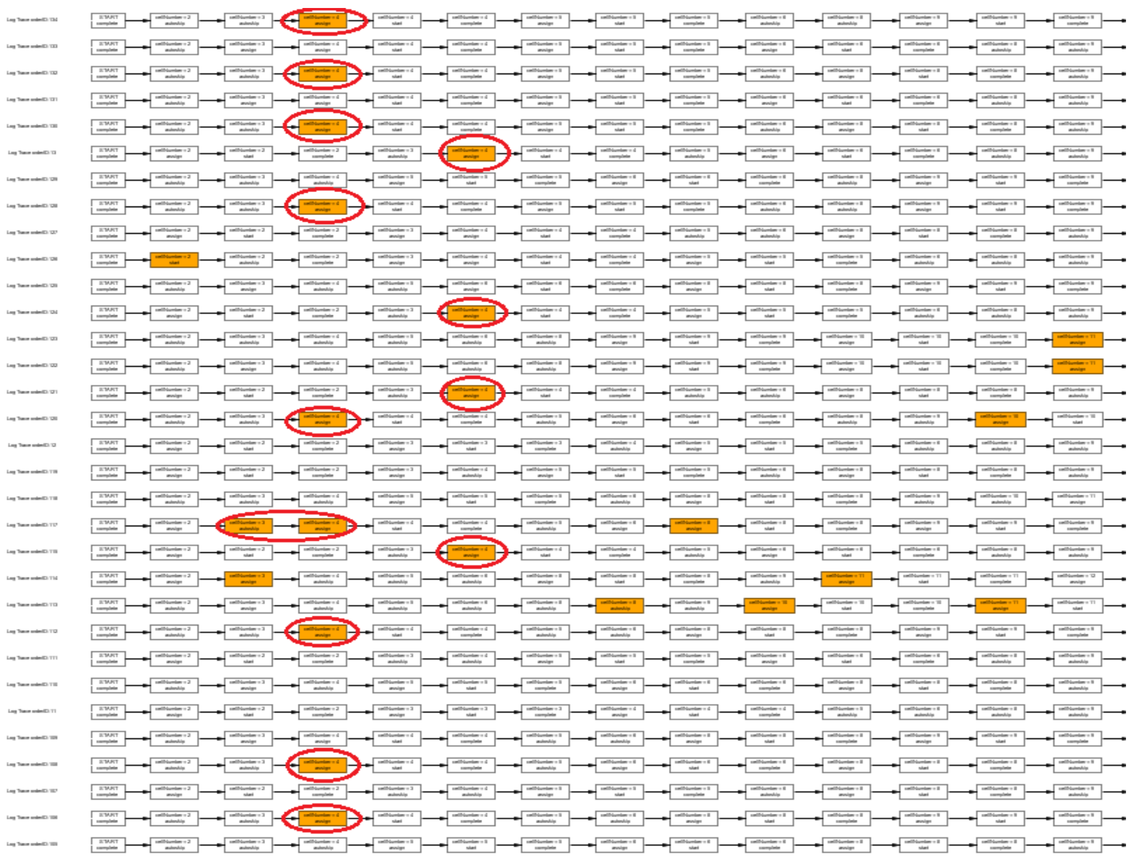


Figure 39. *log view, visualizes each process instance. Incompatible messages related to cell number 4 is highlighted with red colour circles.*

4.2.4.2 Process diagnosis and enhancement

ProM’s *LTL Checker* plug-in verifies conformity of the input event log against a set of rules / requirements defined at design stage that must be fulfilled. *Figure 28* shows examples of informal descriptions of ‘system correctness’.

Figure 40 shows a part of the heuristic model obtained for FASTory. The red highlighted part of the model indicates behaviour considered abnormal after comparison against the rules defining the expected system behaviour. Specifically, Cell 2 is bypassed (autoskip), and then cell number 3 is assigned to work on the product (assign). In FASTory, it is expected that when “assign” notification message for a cell is received, “start” and “complete” notification messages are immediately following. Additionally, it is expected to observe only one output arc from the task cellNumber =3 (assign). However, this part of the model is displaying three output arcs from the task cellNumber3 (assign), while just one of them (assign, start, complete) is acceptable such a problem can be caused by:

1. A need to modify the model, as it is not projecting the real behaviour of processes. This is improbable because the calculated fitness value of the model is high.

2. A need to redefine the rules (defined by CEP) for collecting and aggregating the information (the information system).This is improbable because the problem analyzed occurs only in one part of the model.
3. A need to address communication problems or controller failure in Fastory. This is the most likely situation. Improper communication between some parts of the line and the server may be caused by e.g. high traffic of messages or some component's breakdown.

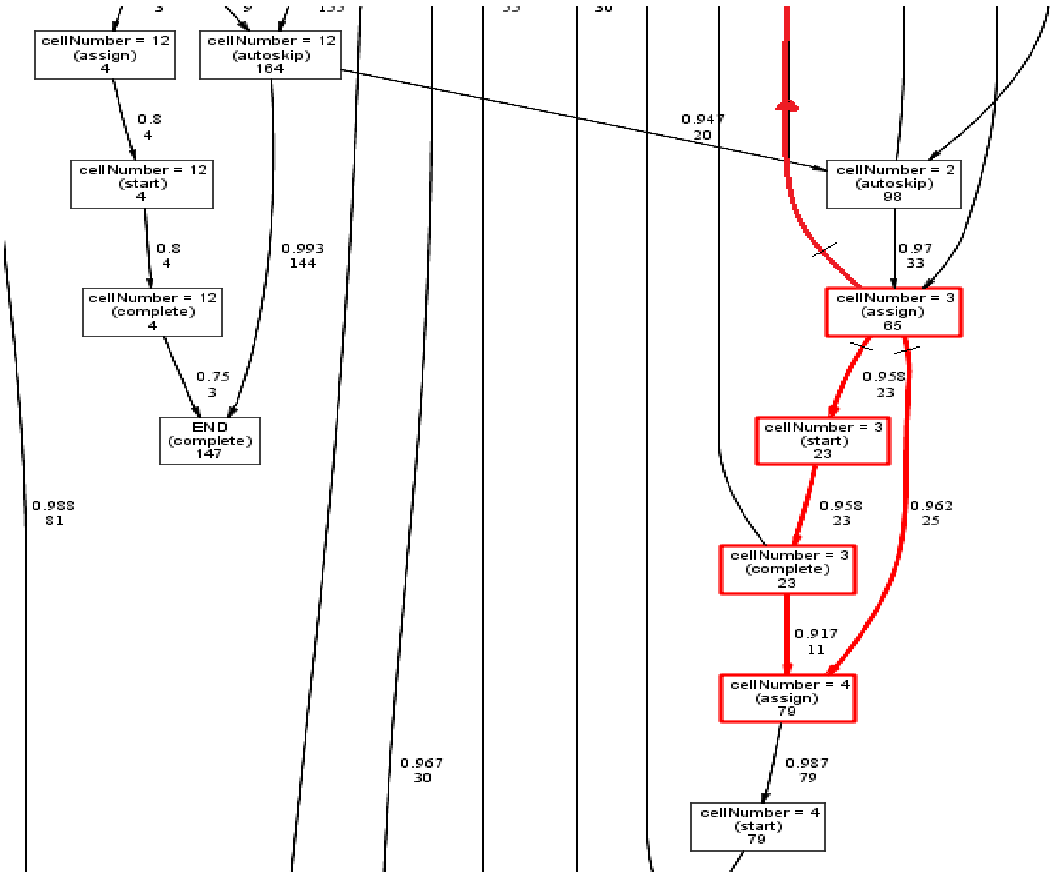

cellNumber = 12 (assign) 4
cellNumber = 12 (autoskip) 164
0.8 4
cellNumber = 12 (start) 4
0.8 4
0.993 144
cellNumber = 12 (complete) 4
0.75 3
END (complete) 147
0.988 81
0.947 20
cellNumber = 2 (autoskip) 98
0.97 33
cellNumber = 3 (assign) 65
0.958 23
cellNumber = 3 (start) 23
0.958 23
0.962 25
cellNumber = 3 (complete) 23
0.917 11
cellNumber = 4 (assign) 79
0.967 30
0.987 79
cellNumber = 4 (start) 79

Figure 40*.Unusual behaviour observed in the generated model from the log*

ProM's LTL checker is used to define the rules that the FASTory line is expected to comply from the behavioural viewpoint. An example of such rule is: "*For_all_activities_always_event_E_implies_eventually_event_F*". Event E is in this case the "assign" event and event F is the "start" event.

The rule is not satisfied for 45 out of 146 process instances. *Figure 41* shows a counterexample for the rule, an incorrect transition between cellNumber 3 and cellNumber4 in the process instance of orderId=100. Two cases are feasible here:

1) Cell 3 performed no operation on the pallet, and the pallet has been bypassed even though the notification message "assign" has been received. 2) The robot of cellNumer 3 has worked on the pallet, but robot-related messages (start, complete) were not submitted by the controller. It

was discovered that in all the other 44 traces, such a wrong transition exists.

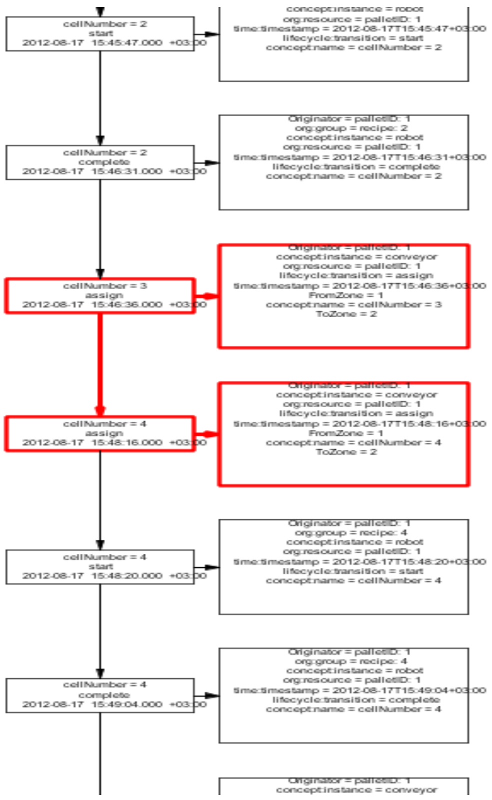
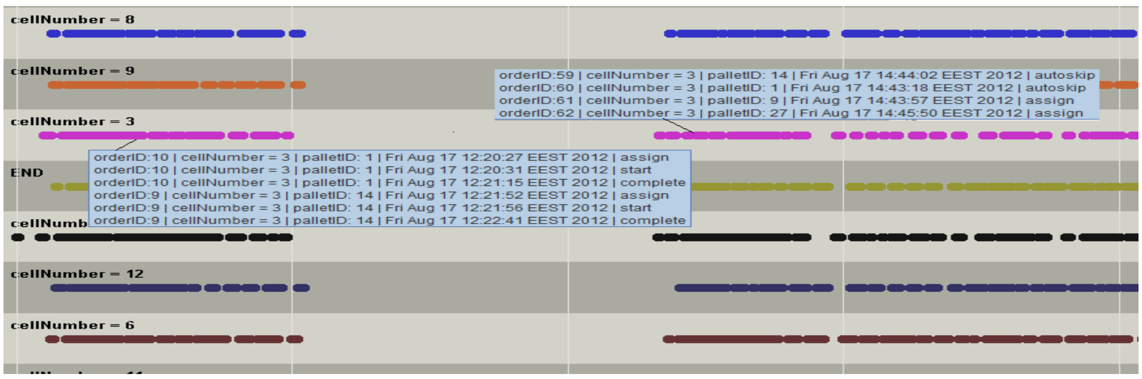

cellNumber = 2
start
2012-08-17 15:45:47.000 +03:00
concept:instance = robot
org:resource = palletID: 1
time:timestamp = 2012-08-17T15:45:47+03:00
lifecycle:transition = start
concept:name = cellNumber = 2
cellNumber = 2
complete
2012-08-17 15:46:31.000 +03:00
Originator = palletID: 1
org:group = recipe: 2
concept:instance = robot
org:resource = palletID: 1
time:timestamp = 2012-08-17T15:46:31+03:00
lifecycle:transition = complete
concept:name = cellNumber = 2
cellNumber = 3
assign
2012-08-17 15:46:36.000 +03:00
Originator = palletID: 1
concept:instance = conveyor
org:resource = palletID: 1
lifecycle:transition = assign
time:timestamp = 2012-08-17T15:46:36+03:00
FromZone = 1
concept:name = cellNumber = 3
ToZone = 2
cellNumber = 4
assign
2012-08-17 15:48:16.000 +03:00
Originator = palletID: 1
concept:instance = conveyor
org:resource = palletID: 1
lifecycle:transition = assign
time:timestamp = 2012-08-17T15:48:16+03:00
FromZone = 1
concept:name = cellNumber = 4
ToZone = 2
cellNumber = 4
start
2012-08-17 15:48:20.000 +03:00
Originator = palletID: 1
org:group = recipe: 4
concept:instance = robot
org:resource = palletID: 1
time:timestamp = 2012-08-17T15:48:20+03:00
lifecycle:transition = start
concept:name = cellNumber = 4
cellNumber = 4
complete
2012-08-17 15:49:04.000 +03:00
Originator = palletID: 1
org:group = recipe: 4
concept:instance = robot
org:resource = palletID: 1
time:timestamp = 2012-08-17T15:49:04+03:00
lifecycle:transition = complete
concept:name = cellNumber = 4
Originator = palletID: 1
concept:instance = conveyor

Figure 41. *Incorrect transition between cell 3 and cell 4 of trace number 100*

Messages may be lost because of malfunctioning controllers or heavy traffic (many controllers sending messages at the same time on the server side). The ProM Dotted Chart plug-in, can be helpful to discover and troubleshoot such problems. The plug-in operates only on event logs (not process models) allowing the monitoring of the number of messages received from different controllers at the same time on the server side / the load of the messages at each time instant.

Figure 42 shows the Dotted Chart Analysis tool in action: for Cell 3, after the time instance of "14:29:03", none of the robot-related messages (start and complete) was sent by robot's controller.



```
orderID:59 | cellNumber = 3 | palletID: 14 | Fri Aug 17 14:44:02 EEST 2012 | autoskip
orderID:60 | cellNumber = 3 | palletID: 1 | Fri Aug 17 14:43:18 EEST 2012 | autoskip
orderID:61 | cellNumber = 3 | palletID: 9 | Fri Aug 17 14:43:57 EEST 2012 | assign
orderID:62 | cellNumber = 3 | palletID: 27 | Fri Aug 17 14:45:50 EEST 2012 | assign
```

```
orderID:10 | cellNumber = 3 | palletID: 1 | Fri Aug 17 12:20:27 EEST 2012 | assign
orderID:10 | cellNumber = 3 | palletID: 1 | Fri Aug 17 12:20:31 EEST 2012 | start
orderID:10 | cellNumber = 3 | palletID: 1 | Fri Aug 17 12:21:15 EEST 2012 | complete
orderID:9 | cellNumber = 3 | palletID: 14 | Fri Aug 17 12:21:52 EEST 2012 | assign
orderID:9 | cellNumber = 3 | palletID: 14 | Fri Aug 17 12:21:56 EEST 2012 | start
orderID:9 | cellNumber = 3 | palletID: 14 | Fri Aug 17 12:22:41 EEST 2012 | complete
```

Figure 42. *Dotted Chart implementation for cellNumber 3 messages*

CONCLUSIONS

The results reported in this thesis include:

- Data mining approaches employed for diagnosis of faults:

  Data mining approaches are employed to characterize the behaviour of pieces of equipment placed in a production line testbed from the viewpoint of energy consumption. During training phase, the energy signature of the system components is associated with semantics concerning the workload of the conveyor belts. At validation phase, real time data coming from the line is input to the classifier and the output obtained is compared against the output of a rule based engine defined offline. Chain consecutive mismatches pinpoint to possible gradual deterioration of expected behaviour. In the presented scenario, such deterioration would translate to a misalignment of conveyor segments.

- Evaluation of the usability of process mining applications for:
  - Discovery of underlying models and performance related issues.
  - Conformance check of a reference model against the original event log. This resulted in highlighting significant information concerning communication failures in-between pieces of equipment

REFERENCES

[1] H.M.W. (Eric) Verbeek, H.M.W. (Eric) Verbeek, ProM 6 Tutorial, August 2010

[2] W.M.P. van der Aalst, A.J.M.M. Weijters, L. Maruster, Workflow mining: discovering process models from event logs, IEEE Trans. Knowl. Data Eng. 16 (9) (2004)1128–1142.

[3] Corina Postelnicu, Navid Khajehzadeh, Jose L. Martinez Lastra, 'Embedded Service Oriented Diagnostics based on Energy Consumption Data', IEEE ICIAfS'12, Factory Automation Systems and Technology Laboratory Tampere University of Technology Tampere, Finland .

[4] Navid Khajehzadeh , Corina Postelnicu, Jose L. Martinez Lastra, 'Detection of abnormal energy patterns pointing to gradual conveyor misalignment in a factory automation testbed',©2012 IEEE, Factory Automation Systems and Technology Laboratory Tampere University of Technology Tampere, Finland, SMC2012

[5] Robert J. Ellison, Andrew P. Moore, 'Trustworthy Refinement Through Intrusion-AwareDesign (TRIAD)', Networked Systems Survivability, October 2002 , Pittsburgh, PA 15213-3890

[6] "Definition of Application Landscape". Software Engineering for Business Information Systems (sebis). Jan 21, 2009. Retrieved January 14, 2011

[7] W.M.P. van der Aalst. Business Process Management Demystified: A Tutorial on Models,Systems and Standards for Workflow Management. In J. Desel, W. Reisig, and G. Rozenberg, editors, Lectures on Concurrency and Petri Nets, volume 3098 of Lecture Notes in Computer Science, pages 1–65. Springer, Berlin, 2004.

[8] M. Weske. Business Process Management: Concepts, Languages, Architectures. Springer, Berlin, 2007.

[9] Staffware Process Suite. http://www.sta_ware.com/

[10] COSA solutions. COSA User Manua l (Pullheim, COSA solutions GmbH, 1996)

[11] Wil M.P. van der Aalst, Process Mining Discovery, Conformance and Enhancement of Business Processes, © Springer-Verlag Berlin Heidelberg 2011

[12] Process Analysis and Organizational Mining in Production Automation Systems Engineering" , Vienna University of Technology, Austria

[13] Fenton, N., Neil, M.: A Critique of Software Defect Prediction Models. IEEE Transactions on Software Engineering 25, 675-689 (1999)

[14] 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)

[15] Mockus, A., Weiss, D., Zhang, P.: Understanding and Predicting Effort in Software Projects. In: International Conference on Software Engineering (ICSE), pp. (2003)

[16] Wikan Danar Sunindyo Thomas Moser Dietmar Winkler Stefan Biffl,"Process Analysis and Organizational Mining in Production Automation Systems Engineering" , Vienna University of Technology, Austria

[17] Minseok Song and Wil M.P. van der Aalst. Supporting Process Mining by Showing Events at a Glance. Eindhoven University of Technology.

[18] ing. J.C.A.M. Buijs: Mapping Data Sources to XES in a Generic Way, Master Thesis, Eindhoven, March 2010

[19] Object Management Group, "Business Process Modeling Notation (BPMN) Specification", Final Adopted Specification, February 2006.

[20] Object Management Group, UML Specification (v1.5), http://www.omg.org/uml, March 2003

[21] A.-W. Scheer, ARIS - Business Process Frameworks (3rdEd), Springer, 1999

[22] W.M.P. van der Aalst, Process-Aware Information Systems: Lessons tobe Learned from Process Mining, Department of Mathematics and Computer Science,Eindhoven University of Technology

[23] Yagang Zhang; Jinfang Zhang; Jing Ma; Zengping Wang, "Fault Detection Based on Data Mining Theory ",IEEE Conference Publications, Intelligent Systems and Applications, 2009. ISA 2009, Page(s): 1 – 4.

[24] J.W. Han, M. Kamber, Data Mining: Concepts and Techniques, Second Edition. San Francisco: Morgan Kaufmann, Elsevier, 2006.

[25] D. Dursun, F. Christie, M. Charles and R. Deepa, "Analysis of healthcare coverage: A data mining approach," Expert Systems with Applications, vol. 36(2), pp. 995-1003, March 2009.

[26] Y.J. Kwon, O.A. Omitaomu and G.N. Wang, "Data mining approaches for modelling complex electronic circuit design activities," Computers &Industrial Engineering, vol.54(2), pp.229-241, March 2008.

[27] Vapnik, VN, Cortes, C.," Support Vector Networks", Machine Learning, vol. 20, pp. 273-297, 1995.ww.tut.fi/triphome/julkai-sut/welcome.html

[28] Process mining website. Accessed September 14th, 2012.

[29] Van Der Aalst, W., et al .Process mining manifesto, Lecture Notes in Business Information Processing 99 LNBIP (PART 1) , pp. 169-194 , 2012

[30] W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 8(1):21–66, 1998.

[31] W.M.P. van der Aalst, K.M. van Hee, A.H.M. terHofstede, N. Sidorova, H.M.W. Verbeek,M. Voorhoeve, and M.T. Wynn. Soundness of Workflow Nets: Classification, Decidability, and Analysis. Formal Aspects of Computing, 2011.

[32] http://www.yawlfoundation.org/ , accessed September 17, 2012.

[33] A.W. Scheer. Business Process Engineering, Reference Models for Industrial Enterprises. Springer, Berlin, 1994.

[34] 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,16(9):1128–1142, 2004.

[35] A.K.A. de Medeiros, W.M.P. van der Aalst, and A.J.M.M. Weijters. Workflow Mining: CurrentStatus and Future Directions. In R. Meersman, Z. Tari, and D.C. Schmidt, editors, On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, volume 2888 of Lecture Notes in Computer Science, pages 389–406. Springer, Berlin, 2003.

[36] L. Wen, W.M.P. van der Aalst, J. Wang, and J. Sun, Mining Process Models with Non-Free Choice Constructs, Data Min. Knowl. Discov., 15-2 (2007),pp. 145-180.

[37] Carmona, J., Cortadella, J., Kishinevsky, M., A region-based algorithm for discovering Petri nets from event logs , Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5240 LNCS , pp. 358-373

[38] Wen, L., Wang, J., Sun, J.,Detecting implicit dependencies between tasks from event logs . Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 3841 LNCS , pp. 591-603

[39] Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L., Yakovlev, A.: Petrify:a tool for manipulating concurrent specifications and synthesis of asynchronous controllers. IEICE Transactions on Information and Systems E80-D (1997) 315–325

[40] A.K.A de Medeiros, A.J.M.M. Weijters, and W.M.P. van der Aalst. Genetic Process Mining: An Experimental Evaluation. Data Mining and Knowledge Discovery, 14(2):245–304, 2007.

[41] A.J.M.M. Weijters and J.T.S. Ribeiro. Flexible Heuristics Miner (FHM). BETA WorkingPaper Series, WP 334, Eindhoven University of Technology, Eindhoven, 2010.

[42] A.J.M.M. Weijters and W.M.P. van der Aalst. Rediscovering Workflow Models from Event-Based Data Using Little Thumb. Integrated Computer-Aided Engineering, 10(2):151–162,2003.

[43] C.W. Günther and W.M.P. van der Aalst. Fuzzy Mining: Adaptive Process SimplificationBased on Multi-Perspective Metrics. In G. Alonso, P. Dadam, and M. Rosemann, editors,International Conference on Business Process Management (BPM 2007), volume 4714 ofLecture Notes in Computer Science, pages 328–343. Springer, Berlin, 2007.

[44] A.W. Biermann and J.A. Feldman. On the Synthesis of Finite-State Machines from Samplesof Their Behavior. IEEE Transaction on Computers, 21:592–597, 1972.

[45] B.F. van Dongenand W.M.P. van der Aalst. Multi-Phase Process Mining: Building Instance Graphs. In P. Atzeni, W. Chu, H. Lu, S. Zhou, and T.W. Ling, editors, International Conference on Conceptual Modeling (ER 2004), volume 3288 of Lecture Notes in Computer Science, pages 362–376. Springer, Berlin, 2004.

[46] W.M.P. van der Aalst, A.H.M. terHofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. Distributed and Parallel Databases, 14(1):5–51, 2003.

[47] 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, 16(9):1128–1142, 2004.

[48] Rozinat, A., &Aalst, W. van der (2006a). Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models. In C. Bussler et al. (Ed.), BPM 2005 Workshops (Workshop on Business Process Intelligence) (Vol. 3812, pp. 163–176). Springer-Verlag, Berlin.

[49] Wil van der Aalst et al , Process Mining Manifesto, "Business Process Management Workshops 2011, Lecture Notes in Business Information Processing, Vol. 99, Springer-Verlag, 2011

[50] MelikeBozkaya, JoostGabriels, Jan Martijnvan der Werf," Process Diagnostics: a Method Based on Process Mining",Information, Process, and Knowledge Management, 2009, 2009 , Page(s): 22 – 27

[51] P. Hornix, "Performance analysis of business processes through process mining," Master's thesis, Technische Universiteit Eindhoven, 2007.

[52] W. van der Aalst, A. Medeiros, and A. Weijters, "Genetic process mining," in ICATPN 2005, ser. LNCS, vol. 3536, 2005, pp. 48 – 69.

[53] Process mining: discovering and improving Spaghetti and Lasagna processes Wil M.P. van der Aalst, Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on Publication Year: 2011 , Page(s): 1 - 7 IEEE Conference Publications

[54] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R.Wirth. CRISPDM 1.0: Step-by-Step Data Mining Guide. www.crisp-dm.org, 2000.

[55] Mager & Pipe in Analyzing Performance Problems (1997)

[56] B.D. Clinton and A. van der Merwe. Management Accounting: Approaches, Techniques, and Management Processes. Cost Management, 20(3):14–22, 2006.

[57] W. van der Aalst, A. Medeiros, and A. Weijters, "Genetic process mining," in ICATPN 2005, ser. LNCS, vol. 3536, 2005, pp. 48 – 69.

[58] C. G¨unther and W. van der Aalst, “Fuzzy mining: Adaptive processs implification based on multi-perspective metrics,” in BPM 2007, ser. LNCS, vol. 4714, 2007, pp. 328 – 343.

[59] J. M. van der Werf, B. van Dongen, C. Hurkens, and A. Serebrenik, “Process discovery using integer linear programming,” in ATPN, ser. LNCS, vol. 5062, 2008, pp. 368 – 387.

[60] A. Rozinat and W. van der Aalst, “Conformance testing: Measuring thefit and appropriateness of event logs and process models,” in BPM 2005Workshops, ser. LNCS, vol. 3812, 2005, pp. 163 – 176.

[61] W. van der Aalst, H. Reijers, and M. Song, “Discovering social networks from event logs,” Computer Supported Cooperative Work, vol. 14, no. 6,pp. 549 – 593, 2005.

[62] M. Song and W. van der Aalst, “Towards comprehensive support for organizational mining,” Technische Universiteit Eindhoven, Tech. Rep.,2007.

[63] Ahmad Mizan and Greg Franks, Automated Performance Model Construction Through Event Log Analysis, Software Testing, Verification and Validation (ICST), 2012 IEEE, Page(s): 636 – 641

[64] G. Franks, D. Petriu, M. Woodside, J. Xu, and P. Tregunno, “Layered bottlenecks and their mitigation,” in Proc. 3rd International Conference on the Quantative Evaluation of Systems (QEST’06), Riverside, CA, USA, Sep. 11–14 2006, pp. 103 – 114.

[65] Yagang Zhang; Jinfang Zhang; Jing Ma; Zengping Wang, “Fault Detection Based on Data Mining Theory “,IEEE Conference Publications, Intelligent Systems and Applications, 2009. ISA 2009, Page(s): 1 – 4.

[66] Venkatasubramanian, V., Rengaswamy, R. K. Yin, and S.N. Kavuri. “Review of process fault detection and diagnosis part iii: Process history based methods.” Computers and Chemical Engineering, vol. 27, pp. 327–346, 2003.

[67] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze,” An Introduction to Information Retrieval” , © 2009 Cambridge University Press , Printed on April 1, 2009

[68] Suykens, J. A. K., &Vandewalle, J. (1999). Least squares support vector machine classifiers. Neural Processing Letters, 9(3), 293–300.

[69] Haifeng W., “Comparison of SVM and LS-SVM for Regression” , IEEE International Conf. On Neural Networks and Brain, vol. 1, pp. 279 –283, 2005.

[70] NaotoshiSeo ,”A Comparison of Multi-class Support Vector Machine Methods for Face Recognition”, Department of Electrical and Computer Engineering The University of Maryland ,December 6, 2007

[71] Process Mining of Test Processes: A Case Study, Process Mining Applied to the Test Process of Wafer Steppers in ASML, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on , Page(s): 474 – 479 , 2009

[72] W.M.P. van der Aalst et al .: Business process mining: An industrial application , Information Systems, Volume 32, Issue 5, July 2007, Pages 713-732.

[73] B.F. van Dongenand W.M.P. van der Aalst. A Meta Model for Process Mining Data. In Conference on Advanced Information Systems Engineering, volume 161, Porto, Portugal, 2005. 10, 11, 83

[74] C. Gunther. Process Mining in Flexible Environments. PhD thesis, Eindhoven University of Technology, 2009. 10, 16, 18

[75] Christian W. Günther , Standard definition , Fluxicon Process Laboratories , XES Version: 1.0 ,November 25, 2009

[76] W. M. P. van der Aalst · V. Rubin ,H. M.W. Verbeek · B. F. van Dongen , E. Kindler · C. W. Günther, ‘Process mining: a two-step approach to balance between underfitting and overfitting’, 25 November 2008, open access at Springerlink.com

[77] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering,47(2):237–267, 2003.

[78] OMRON Corporation. CJM1M Programming Manual, 2006.

[79] Izaguirre M.J.A.G., Lobov A., Lastra J.L.M., "OPC-UA and DPWS interoperability for factory floor monitoring using complex event processing", Industrial Informatics (INDIN), 2011 9th IEEE , International Conference, pp. 205-211, 2011

[80] S1000 User Manual, Inico Technologies Ltd., Available online: http://www.inicotech.com/doc/S1000%20User%20Manual.pdf

[81] W.M.P. van der Aalst. Veri_cation of Workow Nets. In P. Az_ema and G. Balbo, editors, Application and Theory of Petri Nets 1997, volume 1248 of Lecture Notes in Computer Science, pages 407:426. Springer-Verlag, Berlin,1997.

[82] Wang, H., Chai, TY, Ding JL, Brown M.,"Data Driven Fault Diagnosis and Fault Tolerant Control: Some Advances and Possible New Directions", ActaAutomaticaSinica, Vol. 35, No.6, pp. 739–747, June, 2009.

[83] Mobley, "An Introduction to Predictive Maintenance", 2nd edition, pp. 99- 113, 2002.

[84] Ivan M. Delamer , Jose L.Martinez Lastra , book, Factory information systems in electronic production, Tampere University of Technology, [2007], TTY Laitoskokoelma. Virkakäytössä, Staff use only.

[85] http://www.softwareag.com/corporate/products/aris_platform/aris_controlling/aris_process_performance/overview/default.asp , accessed September 26, 2012

[86] http://www.qpr.com/products/qpr-processanalyzer.htm , accessed September 26, 2012

[87] http://fluxicon.com/disco/ , accessed September 26, 2012

[88] http://www.celonis.de/en/discovery-overview.html , accessed September 26, 2012

[89] http://www.celonis.de/en/orchestra-overview.html , accessed September 26, 2012

[90] http://fluxicon.com/nitro/ , accessed September 26, 2012

[91] http://www.stereologic.com/, accesses October 1, 2012

[92] http://fourspark.no/?page_id=11 , accessed October 1 , 2012

[93] www.pallas-athena.com, accessed October 1 , 2012

[94] www.futuratech.nl ,accessed October 1 , 2012

[95] www.fujitsu.com,accessed October 1 , 2012

[96] www.businesscape.no , accessed October 1 , 2012

[97] www.lsi.upc.edu, http://www.lsi.upc.edu/

[98] S. Jablonski and C. Bussler. Workflow Management: Modeling Concepts, Architecture, and Implementation. International Thomson Computer Press, London, 1996.

[99] D. Harel and R. Marelly. Come, Let's Play: Scenario-Based Programming Using LSCs and the Play-Engine. Springer, Berlin, 2003.

[100] H.M.W. Verbeek, J.C.A.M. Buijs, B.F. van Dongen, and W.M.P. van der Aalst. XES, XESame, and ProM6. available online :http://wwwis.win.tue.nl/~wvdaalst/publications/p566.pdf

[101] Luis E. Gonzalez Moctezuma, Jani Jokinen, Corina Postelnicu, Jose L. Martinez Lastra, Retrofitting a Factory Automation System to Address Market Needs and Societal Changes, Industrial Informatics (INDIN),pp. 413-418, 2012 10th IEEE International Conference

[102] L.B. Jack, A.K. Nandi, Comparison of neural networks and support vector machines in condition monitoring application, in: Proceedings of COMADEM 2000, Houston, TX, USA, 2000, pp.721–730.

APPENDIX: MXML VERSUS XES FORMAT OF EVENT LOGS

MXML was developed in 2003 and it has been a common standard applied for exchanging event logs until 2010. MXML was accepted as event log format by ProM(a tool for process mining) . ProMimport was a tool applied for converting databases with different types to MXML event logs formats.MS Access, Aris PPM, CSV, Apache, Adept, PeopleSoft, Subversion, SAP R/3, Protos, CPN Tools, Cognos, and Staffware are a number of databases supported by ProMimport tool. Users are able to add random (optional) elements to the cases in MXML format.However, later it caused numerous ad hoc extensions to MXML event logs and it appeared as a inadequacy for MXML format . This lack of ability to support multiple arbitrary extensions motivated researchers to develop XES event log format. XES format is supported by ProM, Nitro, XESame, and OpenXES. ProM version 6 applies XES and MXML formats of event logs for process mining goals. Nitro(www.fluxicon.com) is a tool provides a user-friendly and simple environment for converting event logs to XES formats. XESame (http://www.processmining.org/xesame/start) is a useful tool applied for generating XES event logs from database tables(In our work, since data are stored in MySQL tables, XESame applied). OpenXES (http://www.xes-standard.org/openxes/start) is an open source java library can be applied for producing XES logs.

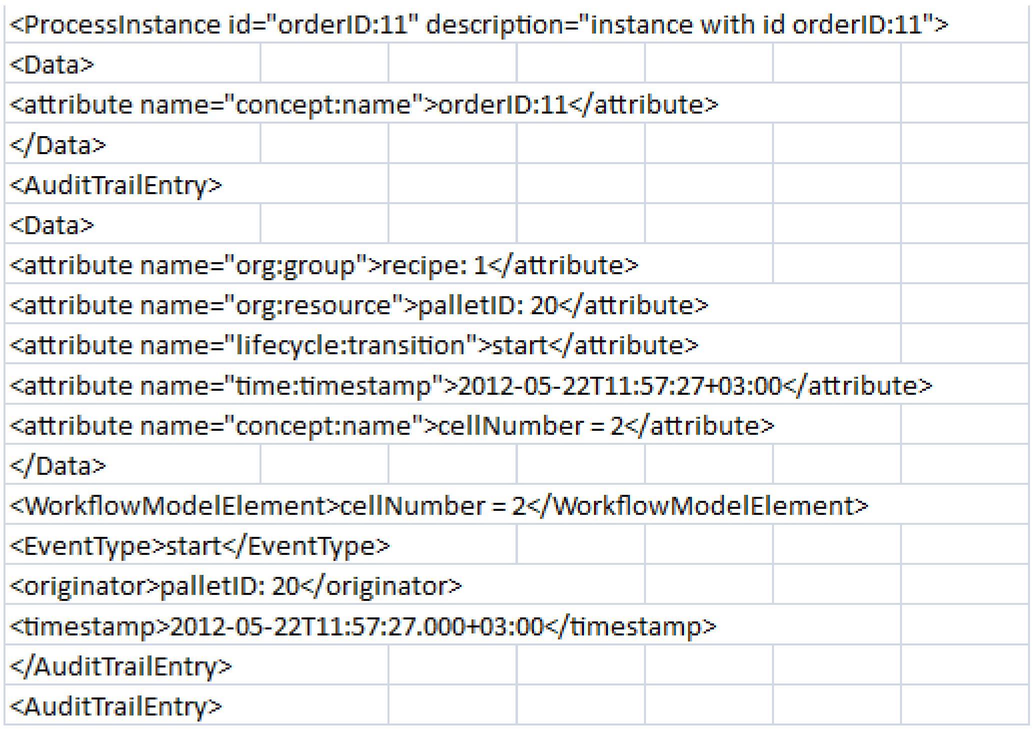
MXML event logs:

MXML is an XML -based user interface markup language. A user interface markup language is a language used for executing and explaining graphical user interfaces and controls. There is no especial and official description for the acronym MXML. However, according to a number of developers it can be an abriviation for “Magic eXtensible Markup Language" .

MXML format [73] of event logs has been an appropriate format for data miners. However, according to [74], MXML has shown some problems at the stage of implementation like false mapping in unstructured environments, difficult translating of logs from processes with complex vertical hierarchy or meaningless semantics of additional attributes stored in the event log. The solution for the raised problems about MXML is discussed and proposed in [18].

Figure 43 shows an MXML format event log. This part of log includes one process instance and one event. Process instances display the cases running in the system. In this example the case is named OrderID 11. Process instances might include more attributes providing more description about each process instance. They also may contain some data values as elements which are useful for comparing the process instances or providing brief and general overviews. AuditTrailEntry is a children element for process instance including the events. Events are stored in AuditTrailEntry as attributes. Events represent the activities in the system. For example, WorkFlowModelElement is showing the name of the activity, in this example it is demonstrating that the event is related to cell number 2. EventType represents the state of the activity e.g. start or complete. The

Originator attribute shows the performer of the activity. As mentioned before, process instances may involve more data values generating more information. In this example ‘group’, ’lifeCycle’, ’timeStamp’ and ‘name’ are additional attributes for ProcessInstance.



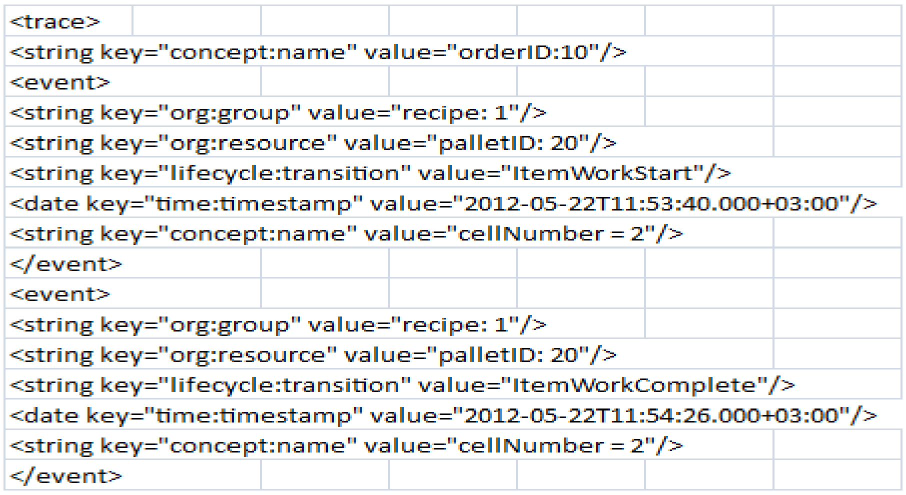
```
<ProcessInstance id="orderID:11" description="instance with id orderID:11">
<Data>
<attribute name="concept:name">orderID:11</attribute>
</Data>
<AuditTrailEntry>
<Data>
<attribute name="org:group">recipe: 1</attribute>
<attribute name="org:resource">palletID: 20</attribute>
<attribute name="lifecycle:transition">start</attribute>
<attribute name="time:timestamp">2012-05-22T11:57:27+03:00</attribute>
<attribute name="concept:name">cellNumber = 2</attribute>
</Data>
<WorkflowModelElement>cellNumber = 2</WorkflowModelElement>
<EventType>start</EventType>
<originator>palletID: 20</originator>
<timestamp>2012-05-22T11:57:27.000+03:00</timestamp>
</AuditTrailEntry>
<AuditTrailEntry>
```

Figure 43. *MXML log format*

EXtensible Event Stream (XES) event logs:

A standard that could also be used to store event logs from various types of information systems is under development by the IEEE Task Force Process Mining (See http://www.win.tue.nl/ieeetfpm). This new event log format is named XES and stands for eXtensible Event Stream. This new format has much more flexibility and solves many problems exist at converting records to MXML format event logs

Figure 44 shows the XES format of the same log shown in MXML format before.



```
<trace>
<string key="concept:name" value="orderID:10"/>
<event>
<string key="org:group" value="recipe: 1"/>
<string key="org:resource" value="palletID: 20"/>
<string key="lifecycle:transition" value="ItemWorkStart"/>
<date key="time:timestamp" value="2012-05-22T11:53:40.000+03:00"/>
<string key="concept:name" value="cellNumber = 2"/>
</event>
<event>
<string key="org:group" value="recipe: 1"/>
<string key="org:resource" value="palletID: 20"/>
<string key="lifecycle:transition" value="ItemWorkComplete"/>
<date key="time:timestamp" value="2012-05-22T11:54:26.000+03:00"/>
<string key="concept:name" value="cellNumber = 2"/>
</event>
```

Figure 44 *. XES log format*

There is no globally defined attribute for XES event logs, and it causes the attributes of the elements inside XES format of event logs to have vague semantics. According to [18], this vagueness is solved by defining some extensions in XES format. Those extensions produce some attributes at different levels in the XES log architecture and they provide some references for the purpose of the interpretation of attributes. Christian W. Günther in [75] defines a meta-model for XES format of event logs. Global attributes are defined as external extensions in order to overcome the problem of vague semantics when further information are required while they are not defined in the context of standard attributes.

In XES log format each *trace* element stands for *ProcessInstance* in MXML log format. It includes events (activities) happening in every trace. *Figure 44* shows that each trace may contain some attributes. In this example the trace includes one attribute shown as ‘name’. The ‘name’ attribute is specified by “concept” standard extension of XES. The ‘concept’ extension provides a specific name for elements of all logs, traces and events. ‘transition’ attribute is defined by “lifecycle” extension. ‘transition’ is referred to the ‘ event type ‘ attribute in MXML format. Usually, ‘transition’ attribute announces the start or complete state of each activity. The other extension shown in *Figure 44* is “org” stands for organisational extension. “org” includes the resource attribute showing the performer of the activity. In order to store the time and date of each activity, ”time” extension is applied. It is important to know that, it is not necessary to define extension for all attributes. It is also worth mentioning that, attributes may involve other attributes. *Figure 44* demonstrates trace or processInstance number 10, including 2 events.