Institut für Parallele und Verteilte Systeme
Abteilung Anwendersoftware

Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Masterarbeit Nr. 3402

A Process Insight Repository supporting Process Optimization
Ein Process Insight Repository zur Unterstützung der Prozessoptimierung

Andrey Vetlugin

Studiengang:	Infotech
Prüfer:	PD Dr. rer. nat. habil. Holger Schwarz
Betreuer:	M. Sc. Christoph Gröger
begonnen am:	01.06.2012
beendet am:	20.12.2012
CR-Klassifikation:	H.2.8, J.1

Erklärung

Ich versichere, Diese Arbeit selbstständig verfasst zu haben.

Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet.

Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens.

Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Unterschrift:

Stuttgart, 20.12.2012

Table of Contents

List of Tables

List of Figures

Abstract

Existing solutions for analysis and optimization of manufacturing processes, such as online analysis processing or statistical calculations, have shortcomings that limit continuous process improvements. In particular, they lack means of storing and integrating the results of analysis. This makes the valuable information that can be used for process optimizations used only once and then disposed.

The goal of the Advanced Manufacturing Analytics (AdMA) research project is to design an integrated platform for data-driven analysis and optimization of manufacturing processes using analytical techniques, especially data mining, in order to carry out continuous improvement of production. The achievement of this goal is based on the integration of the data related to the manufacturing processes, especially from Manufacturing Execution Systems (MES), with the other operating data, e.g. from Enterprise Resource Planning (ERP) systems.

This work is based on AdMA platform described in [1] and Deep Business Process Optimization platform described in [2]. It is focused on the conceptual development of the Process Insight Repository, which is a part of the AdMA platform. The Process Insight Repository is aimed at storing the manufacturing process related data and the insights associated with it. Being part of the AdMA platform, the Process Insight Repository is oriented on storing the insights retrieved by application of data mining techniques to the data of manufacturing processes, so that the newly extracted knowledge can be stored along with the process data itself.

Chapter 2 describes the conceptual schema of the Process Insight Repository. The conceptual schema defines what data must be stored in the Process Insight Repository and how different parts of this data are interconnected.

Chapter 3 provides a review of technologies that can be used for the implementation of the Process Insight Repository. This includes technologies for storing manufacturing process data, free form knowledge and data mining related data.

Chapter 4 describes the details of the prototype implementation of the Process Insight Repository.

The result of this work is the created conceptual schema of the Process Insight Repository and a prototype implementation as a proof of concept.

1 Introduction

Existing solutions for analysis and optimization of manufacturing processes, such as online analysis processing or statistical calculations, have shortcomings that limit continuous process improvements. In particular, they lack means of storing and integrating the results of analysis. This makes the valuable information that can be used for process optimizations used only once and then disposed.

The goal of this work is to develop the general concept of the Process Insight Repository. This repository must provide storage for all the available manufacturing process data and the insights into these processes. The Process Insight Repository is a part of the Advanced Manufacturing Analytics (AdMA) platform that helps its users to extract knowledge from the accumulated process data, store this knowledge and use it for optimization of manufacturing processes.

The goal of the Advanced Manufacturing Analytics (AdMA) research project is to design an integrated platform for data-driven analysis and optimization of manufacturing processes using analytical techniques, especially data mining, in order to carry out continuous improvement of production (see Figure 1). A part of the AdMA platform that is dedicated to storing the insights into manufacturing processes retrieved by various analytical techniques is the Process Insight Repository.

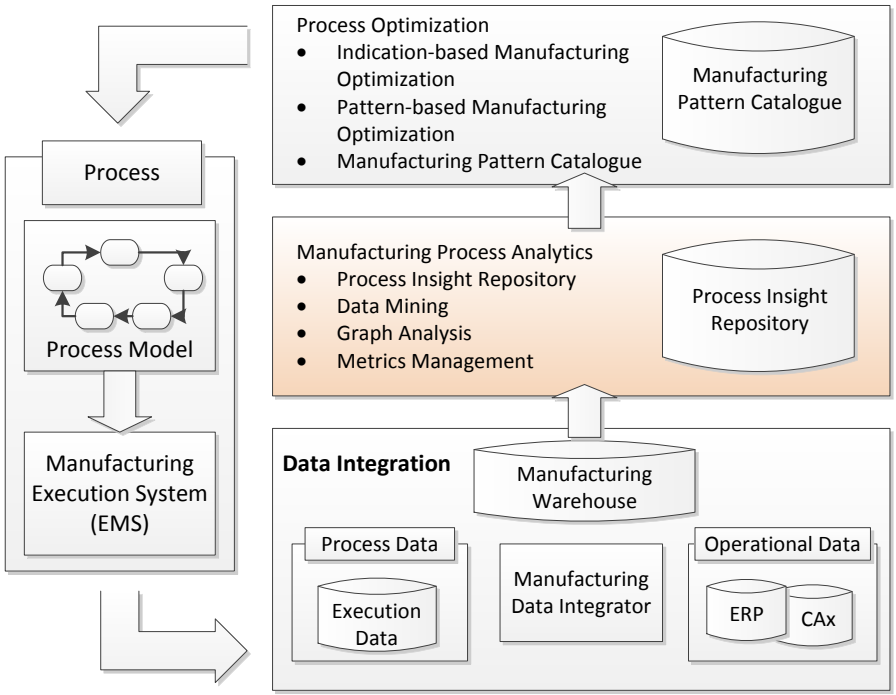

Figure 1: The Process Insight Repository in the context of AdMA platform

This work develops the idea of the Process Insight Repository that was introduced as part of the AdMA platform described in [1] and part of the Deep Business Process Optimization platform

described in [2]. The Process Insight Repository is aimed at storing the manufacturing process related data and the insights associated with it. Being part of the AdMA platform, the Process Insight Repository is oriented on storing the insights retrieved by application of data mining techniques to the data of manufacturing processes, so that the newly extracted knowledge can be stored along with the process data itself and reused later.

In order to achieve the specified goal, several tasks were accomplished in this work. The first task was to design a conceptual schema of the Process Insight Repository. In this task it was defined what kinds of data need to be stored, how different parts of the schema are interconnected. The second task was to define which technologies and products can be used for the implementation of the Process Insight Repository and how they can be used; which technologies suit the needs of the repository the best. The third task was to implement a prototype of the Process Insight Repository as a proof of concept and in order to see challenges that arise in such implementation.

1.1. Problem description

There is a set of problems in optimization of manufacturing processes which can be solved by application of data mining techniques. Imagine that to produce some product, a part must be processed on a production line consisting of a number of machines. Each machine transforms this part; some machines can use external materials. For the same manufacturing operation different production lines can use different types of machines, different materials coming from different vendors. On the quality control of the ready-made product, it turns out that one line has more defective parts than another. What is the reason that some of the produced parts have defects? Which factors influence final characteristics of the product? Is it because of old machines that are used, different material quality or insufficient staff qualification? This is an example of questions that can be answered using data mining techniques. In this case, a decision tree can be built in order to perform the root cause analysis.

With the current state of manufacturing technologies, a lot of data about the performance of a manufacturing process can be accumulated. Valuable knowledge can be extracted from this data using data mining techniques, but performing data mining tasks requires special knowledge. However, if we consider the field of manufacturing process optimization, there are some common patterns that can be applied to many processes. One of these patterns, root cause analysis, is given in the example above.

1.2. Use cases of the Process Insight Repository

The PIR is a part of the platform that does not have any graphical user interfaces. It provides an API to the higher level of the platform which is applications that use the PIR as a service provider and provide different kinds of user interfaces to the end users of the platform. User interface applications provide the functionality by performing API calls to the PIR on user request; they can run on various hardware with different characteristics, e.g. on a tablet pc, a laptop or a personal computer. There are three types of users that work with the Process Insight Repository: non-expert user, advanced user and data mining specialist. An example of a non-expert user can be a shop floor worker who accesses the Process Insight Repository with a

portable table computer. An advanced user can be a production supervisor, who has more advanced knowledge and can use extended functionality of the Process Insight Repository. A data mining specialist can be a business intelligence analyst, who is able to perform analysis of manufacturing processes using analytical software.

Any type of users must be able to navigate the data that is available in the PIR. This includes navigating process model data (build-time process model) and process execution data (run-time process model): see available processes and process instances, inspect the structure of a particular process – production steps performed within a process, machines and materials being used. A user must be able to see and inspect all the insights that are available for a selected entity of the process model. These user activities are shown in a use case diagram in Figure 2.

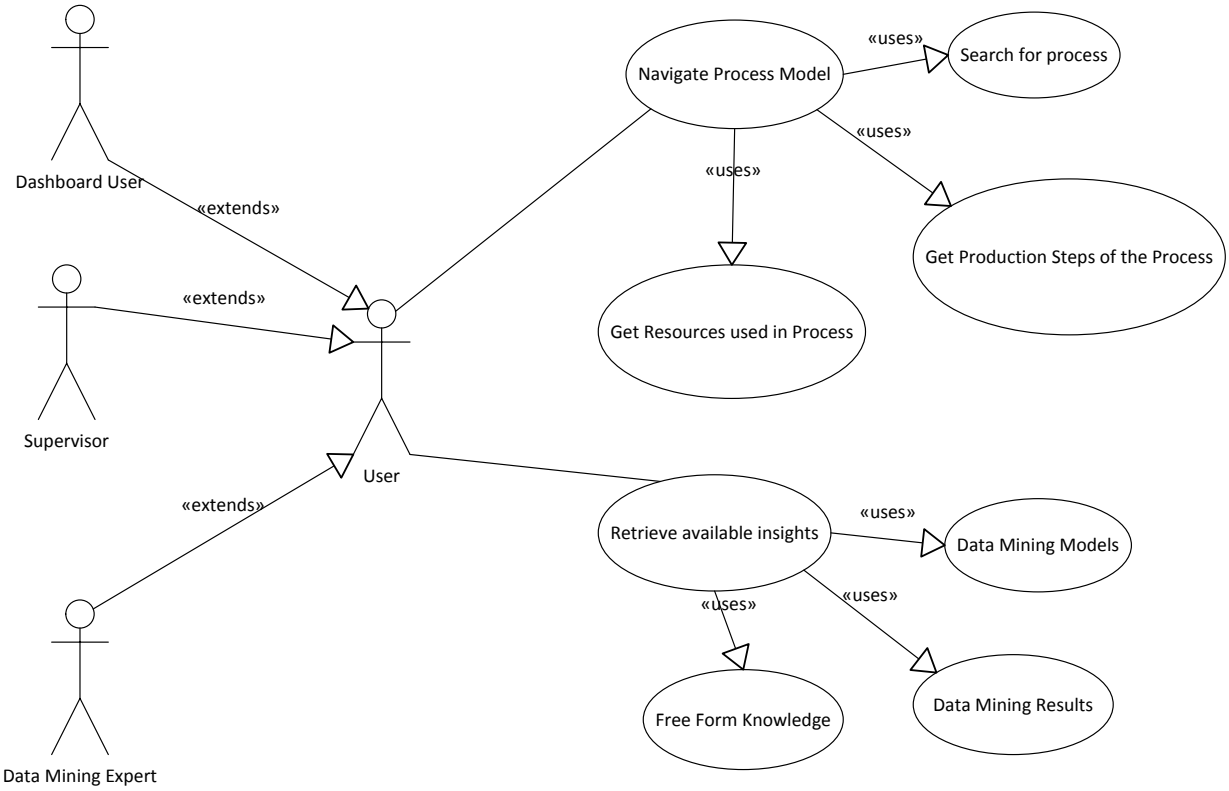

Figure 2: Use case diagram: navigation of the Process Insights Repository

Non-expert user is a user who does not have any specific knowledge about data mining techniques. Non-expert user accesses the PIR via a user interface application. This user must be able to execute the predefined data mining tasks and inspect the results. The data mining tasks must be prepared in such a way, that the user must not specify any data mining specific parameters. This means that the data mining tasks must not require any parameters at all or have some predefined reasonable default parameters. Several types of non-expert users can be defined, with different subsets of data mining tasks available to each type of non-expert users. Figure 3 shows the two predefined data mining tasks that can be executed by non-expert users.

Advanced user performs the same tasks with the platform as non-expert user, but being an advanced user he has an extended user interface to the platform. For example, advanced user can alter some parameters of the data mining tasks in order to receive improved data mining results (see Figure 3).

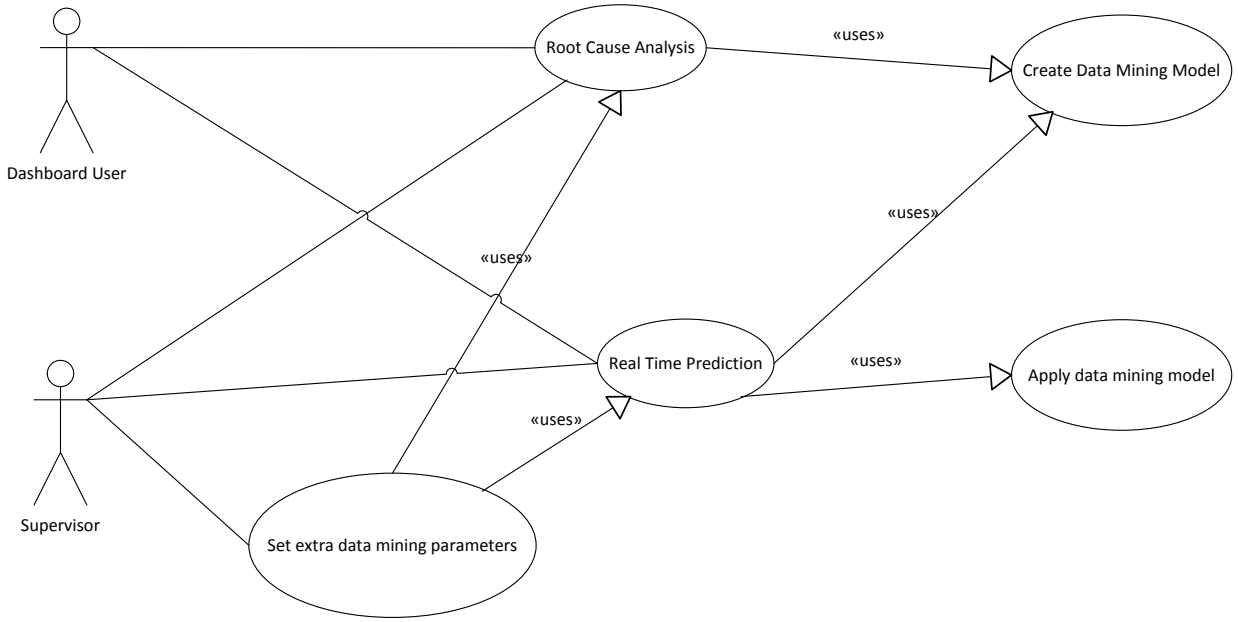

Figure 3: Use case diagram: predefined data mining tasks for users

Two sample data mining use cases were selected for the prototype implementation of the Process Insight Repository:

- Root Cause Analysis for a selected process metric. This use case is based on building a decision tree for a process metrics selected by the user in order to find attributes of the manufacturing process that influence the selected metric. This use case is described in [3].
- Real Time Prediction for a metric of an uncompleted process. This use case is based on classification or regression data mining techniques in order to predict a metric of an uncompleted manufacturing process based on the available data of completed processes in the Process Insight Repository. This use case is described in [3].

Sample use cases “Root Cause Analysis” and “Real Time Prediction” assume that the PIR must provide an API to the higher level of the platform to perform these tasks. As far as users do not have any specific knowledge in data mining, the GUI that is using the API provided to the user interface must be simple and must not require the user to provide any specific settings or parameters. If the data mining process requires any parameters, they must be predefined and stored as settings of the data mining process in order to exempt the user from specifying these parameters. It is the task of the PIR to perform all the steps of the data mining workflow within a simple API call from the user interface application.

Data mining specialist is a user who has knowledge in data mining and is therefore able to perform data mining of the PIR data using external data mining tools. This user can create new data mining tasks and make them available for other users, so that these new data mining tasks can be easily accessed with the standard interface of the PIR (see Figure 4). As far as data mining specialist will use external data mining tools, the PIR can assist by supporting selected steps of the data mining process (see Section 3.2.1 for the description of these steps), i.e. data preparation. For example, data mining specialist can navigate the PIR and select a subset of data that he is interested in (e.g., a particular manufacturing process); the PIR can do the data transformation to prepare the selected data for data mining analysis with external data mining tool and create a view on the selected data or provide a ready-made query to access the data from the data mining tool.

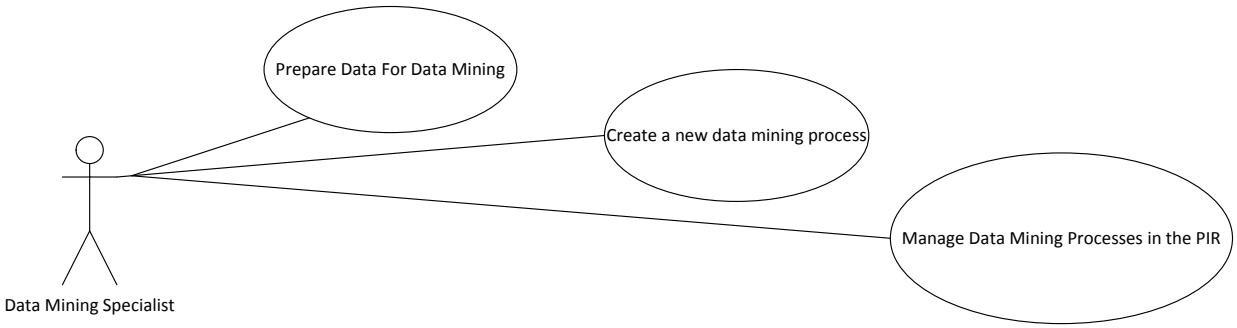

Figure 4: Use case diagram: data mining specialist

2 Conceptual Schema of the Process Insight Repository

The main task of the Process Insight Repository is to store the holistic information about manufacturing processes executed in a factory. This includes all the available process data itself and all the insights that are associated with a manufacturing process. This information must be well structured in order to be easily accessible and useful.

A conceptual schema of the Process Insight Repository is designed in this chapter. The discussion covers the kinds of manufacturing process data that must be stored in the Process Insight Repository, kinds of process insights that can be stored, and how different parts of the process model are related to each other and associated with process insights. As far as some parts of the schema can be modeled in different ways, different schema variants are discussed. A variant of the complete conceptual schema of the Process Insight Repository is given in the end of this chapter.

2.1. Manufacturing process model

The important part of the Process Insight Repository is the model of the manufacturing process itself. All the relevant data for process analysis must be stored and made available. The process model can be viewed from two perspectives.

One perspective is the build-time model of a manufacturing process and represents the manufacturing process as it is modeled by a process engineer. The build-time perspective describes a manufacturing process of a particular product group, i.e. production steps that must be carried out, types of machines used for that, materials used in the process, etc. The build-time model contains all the production steps that can be executed within a process, even if some production steps are planned to be executed only under some conditions (e.g., rework of a part in case of failing quality control). If a process has production steps that are executed in parallel, or different conditional branches of production steps, all this information is part of the build-time process model.

Another perspective is the run-time perspective of a manufacturing process. It describes the factual execution of the process on the real equipment with real materials. The run-time process model contains multiple process instances of one modeled process. Each process instance in the run-time model describes the production steps that were actually fulfilled during execution, the particular machines that were used for execution of production steps, materials that were used in production (which batch, from which vendor), the performance measurements of process execution, etc. In contrast to the build-time model, run-time model contains only the production steps that were in fact executed, i.e. the run-time model does not contain branches of production steps that were not executed under some conditions.

The remaining part of this chapter presents entities of the run-time process model and corresponding entities of the build-time process model.

2.1.1. Support for process versioning

The highest level of the manufacturing process model is the Manufacturing Process itself in the build-time model and Manufacturing Process Instance in the run-time model (see Figure 5). They consist of Production Steps and Production Step Instances respectively. A Manufacturing Process is engineered to produce a particular Product Group. The description of these entities is given above.

A Manufacturing Process Instance is the execution of the corresponding Manufacturing Process and is initiated by a Production Order. A Production Order is an order to produce a Product of a particular Product Group that comes from a Customer. It holds information, such as amount of goods to be produced and priority of the order. [4]

A Customer can be categorized as an External or Internal customer. An External Customer of an organization is a customer who is not directly connected to that organization. An Internal Customer is a customer who is directly connected to an organization and is usually internal to the organization.

A Product can be defined as an End Product if it is a product that is ordered by a customer or as Part if it is an intermediate product that will be used as an input material later in another Manufacturing Process.

It is common that the process can be changed with time. The Process Insight Repository must properly handle these changes and provide support for the user to identify different processes and variants of the same process. [5] In order to do this, two entities were introduced into the build-time model of the process: Process Variant and Process Version (see Figure 5). These entities allow the user to identify variants and versions of the same process and retain a detailed account of the changes in the process.

Process Variant is a variant of a Manufacturing Process. For example, if Manufacturing Process can use one of the two different materials, but each material requires different sequence of production steps, this can be modeled as two Process Variants of the same Manufacturing Process.

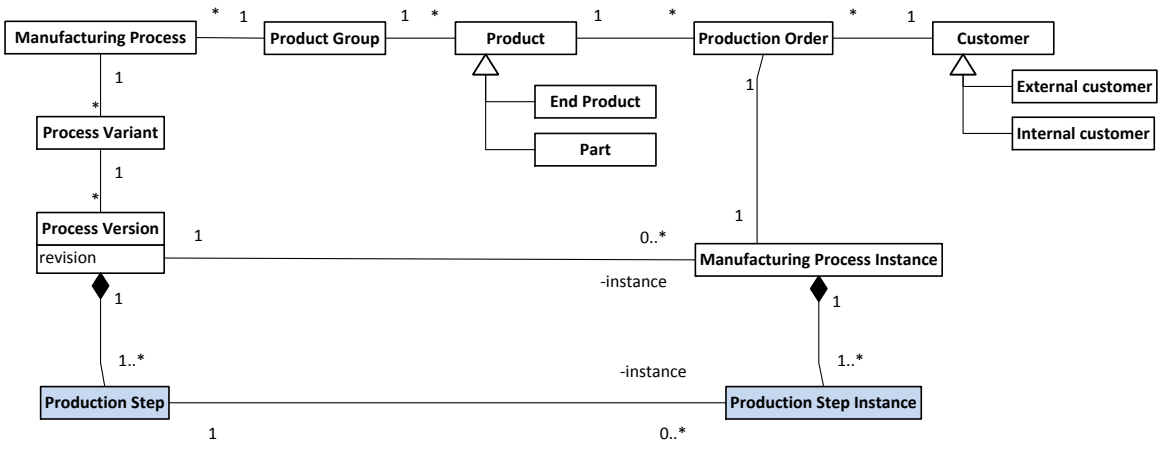

Figure 5: Process Versions, Product and Production Step

With the lapse of time a Process Variant can change. For example, some production steps can be replaced with other steps, new production steps can be added to the manufacturing process or the order of execution can change. These changes can be slight and still represent the same Process Variant. Hence, each such version of a Process Variant is stored as a Process Version. This versioning of the process allows retaining of historical versions of the process and makes possible to compare the current state of the manufacturing process with the previous historic versions in order to reveal improvements or deteriorations of the process.

2.1.2. Production Step

A Process Version consists of multiple Production Steps – activities that must be performed within a manufacturing process in order to manufacture the Product. Production Steps are the main source of facts of a manufacturing process, they hold information, such as step duration, resources used in production, occurred faults, etc.

Types of Production Steps

There are different ways to classify Production Steps, e.g. main criteria according to [6], is whether a Production Step adds value to the product or not. This classification is useful in terms of manufacturing process optimization as it allows easily finding out optimization targets. However, this classification requires deep analysis of the manufacturing process in advance, reducing the information gain that could be received from application of data mining techniques to the process data. In addition, such classification would separate Production Steps of the same nature, e.g. transportation of goods on a factory premises does not add value, while transportation of goods to the customer is a value-adding activity, although both of the transportation activities would have about the same set of attributes. Hence, such approach is not of much use for the Process Insight Repository.

Instead of this, another approach was selected, that classifies Production Steps by similarity of their properties and hereby is better in terms of storing accumulated process data. This approach allows storing Production Steps with similar attributes in a common structure of the Process Insight Repository. Hence, a set of activity types was selected that is typical in manufacturing and the following types of Production Steps were defined in the Process Insight Repository: Manufacturing Step, Assembly Step, Transportation Step, Testing Step and Misc Step (see Figure 6). [7] [8]

Manufacturing Step is an activity that alters the form and properties of the processed part. Examples of such Manufacturing Steps are casting, molding, forming and machining.

Assembly Step is an activity that involves assembly of different parts or mounting of additional parts on the main processed part.

Testing Step is an activity to perform quality control of the produced parts. The outcome of a Testing Step can affect the execution sequence of a Manufacturing Process Instance, e.g. if a part was tested as defective, it might require additional processing, rework or transportation to the waste storage.

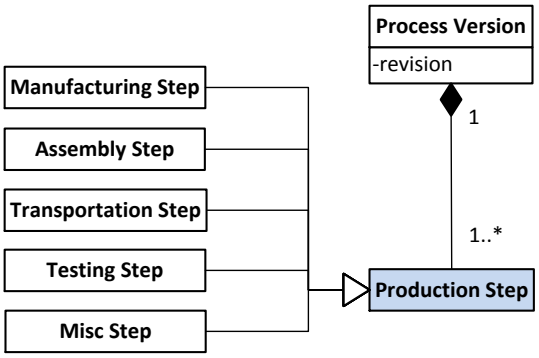

Figure 6: Types of Production Steps

Transportation Step represents various transporting of parts, e.g. using a belt line, elevator or other means.

Misc Step covers the rest of miscellaneous activities that cannot be covered by the above mentioned predefined types of Production Step, e.g. storage of materials, delays, information and other activities. [7]

Physical and geographical location

A Production Step Instance is executed in a particular Work Unit, which has a particular physical geographical location. A Work Unit hierarchically belongs to a Production Line, Area and Factory (see Figure 7). [9]

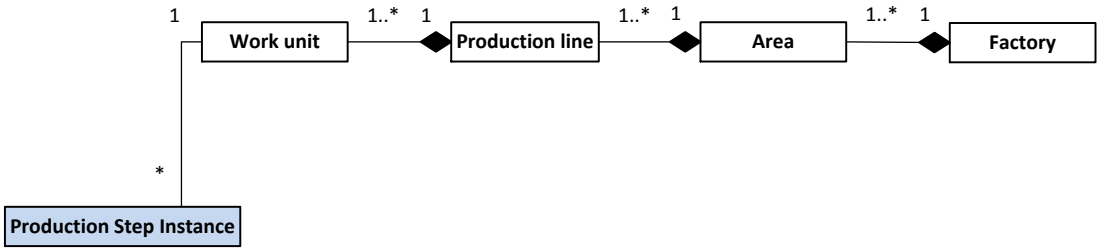

Figure 7: Physical and Geographical location of Production Step Instance

Operating Resource

A Production Step Instance can have Operating Resources associated with it. Two types of Operating Resources were taken for the conceptual schema of the Process Insight Repository: Machine and Production Aid (see Figure 8).

A Machine represents a particular stationary technical manufacturing device. A Machine can have properties, such as machine manufacturer, type, release date, last service and maintenance date, energy consumption and others. Machines are logically grouped into Machine Groups. [4]

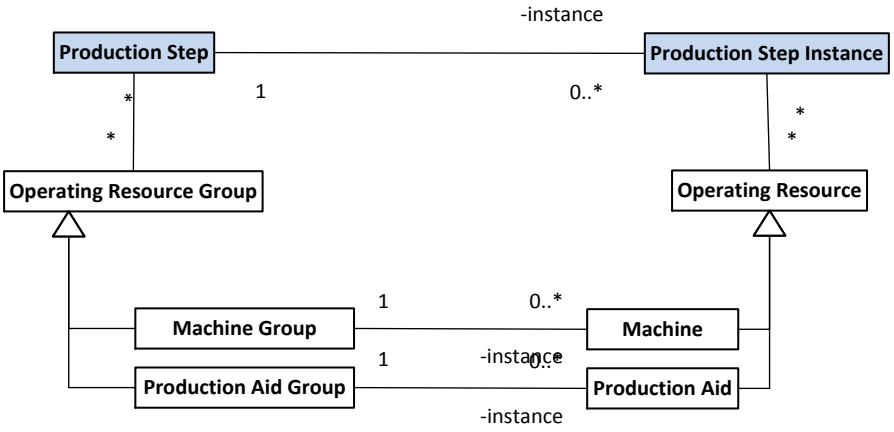

Figure 8: Operating Resources

A Production Aid represents any production means that can be directly used in a Production Step Instance, e.g. tools, facilities, measuring devices, reference manuals, etc. Production Aid is mobile in contrary to Machines. Production Aid is grouped to Production Aid Group.

Employee

Employee is another dimension describing human resources that take part in execution of a Production Step Instance (see Figure 9). Employee dimension can be alternatively viewed as a part of Operating Resources dimension. However, in this work it was allocated as a separate class, because human resources have different attributes compared to machinery resources and are traditionally considered separate from machinery resources in manufacturing. An Employee can be characterized with such attributes as qualification and working experience. Employees are associated with an Employee Group, which is assigned to a Department.

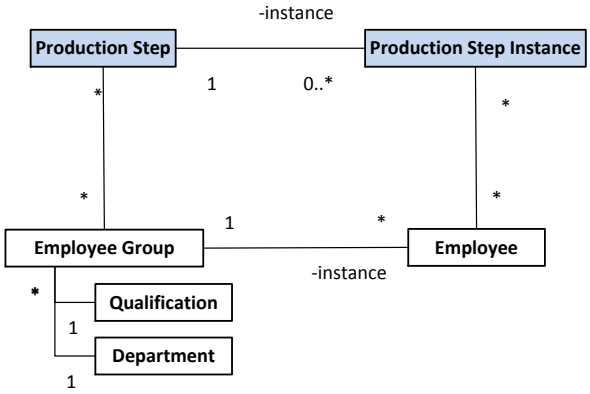

Figure 9: Employee

External Input Material

The External Input Material entity represents input materials that are used in a Production Step. These can be workpieces, raw materials or auxiliary matters. External Input Material Class describes the type of material used in a Production Step, while External Input Material specifies the amount of material used from a Batch (see Figure 10). The Batch represents a material batch that is associated with the exact Material entity and comes from a particular Vendor. No

distinction is made in the conceptual model between External Input Materials that are produced locally and those that are purchased from external companies. When External Input Material is produced on the same factory, the Vendor entity would be the factory itself. [4]

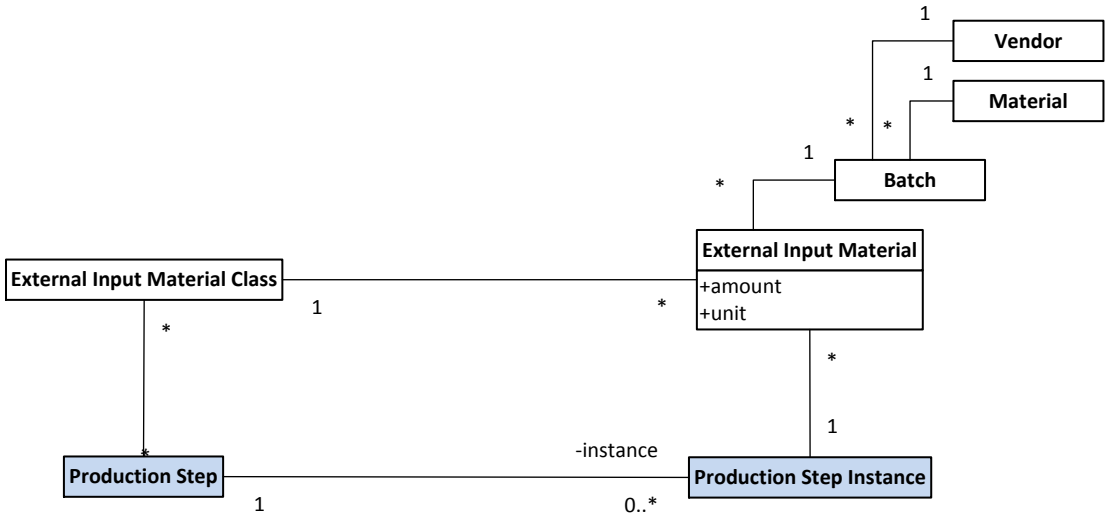

Figure 10: External Input Material

External Input Materials do not represent parts that come from the previous Production Step, i.e. parts that are the main processing target of the Manufacturing Process Instance and are passed from one Production Step to another. These parts are modeled using Material Gateways (see Section 2.1.3).

Operational Material, Emission and Faults

Operational Materials, Emissions and Faults are not modeled at build-time. They exist only as a result of process execution in the run-time model of the process.

Operational Material Consumption represents operational materials that are used in Production Steps but do not go into the manufactured Product. Operational Material Consumption specifies the amount and units of an Operational Material that was consumed in a Production Step Instance (see Figure 11). Operational Materials are grouped into an Operational Material Group, e.g. high tension electricity and low tension electricity are Operational Materials of the group electricity. [4]

Emission Formation specifies the amount and units of various Emissions that take place during the execution of a Production Step Instance. Emissions can be grouped into an Emission Group (see Figure 11). [4]

Fault entity represents faults that happened during the execution of a Production Step Instance. Each Fault belongs to a Fault Group (see Figure 11).

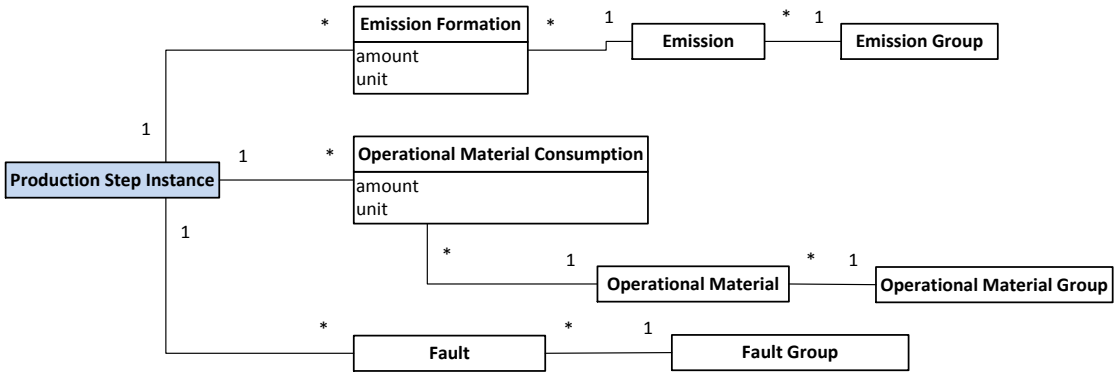

Figure 11: Operational Material, Emission and Fault

2.1.3. Material Gateway

The previous section described the most important part of the manufacturing process model – a Production Step. A term Material Gateway is used for the part of the process model that connects Production Steps with each other. Material Gateways play two main roles in the model of a manufacturing process – they show the execution sequence of Production Steps and the flow of material between Production Steps. This section introduces two solutions that can be used to model Material Gateways in the Process Insight Repository.

Material Gateway option 1: control flow only

A Manufacturing Process consists of Production Steps which are executed and applied to a processed part in a particular order. It can be represented as a directed graph, where nodes are Production Steps and ribs show the execution order of Production Steps. This relationship between Production Steps can be resolved by introducing a recursive predecessor-follower association (see Figure 12). This association has many-to-many multiplicity, because a Production Step can have multiple predecessors and multiple followers.

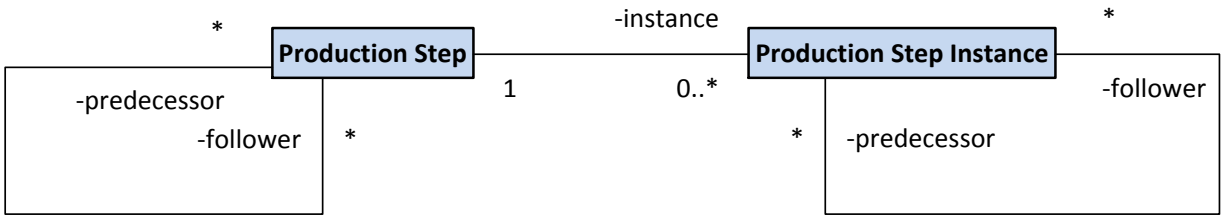

Figure 12: Material Gateway: control flow only

The build-time process model contains all the possible transitions between Production Steps showing the designed graph of a Manufacturing Process. This graph can have loops and conditional branches. However, different relations between Production Steps are modeled identically, e.g. no difference is made if a Production Step is followed by two parallel Production Steps or by one of the two conditional Production Steps.

The run-time process model contains only the transitions between Production Steps that were actually taken during the factual execution of a Manufacturing Process Instance. This means, that the run-time model of a process can have repetitive Production Steps represented by different Production Step Instances or have some of the Production Steps omitted if their execution was conditional and the condition was not met.

This solution is simple to implement. However, it shows only the control flow between Production Steps and does not model the material flow. Hence, this option can be taken for implementation only if the material flow data is not available for some reason or it does not carry much information that can be used for the analysis of the process.

Material Gateway option 2: control flow and material flow

The Material Gateway described in the previous section provides a simple solution, but does not model material flows between Production Steps. If information about material flows is available for the Process Insight Repository, it is important to model it, since this information can be used to analyze the generation of added value to the product by process optimization methods such as Value Stream Mapping. [6]

In order to model the material flow, let us refer to process modeling standards. A manufacturing process is in general similar to a business process in the sense that it also consists of a set of activities interconnected with each other. The main difference is that in a business process activities interact with each other by sending information in a form of data and event signals, while activities in a manufacturing process (Production Steps) transfer physical objects to each other.

An example of a popular business process modeling standard is BPMN – Business Process Modeling Notation. It uses a concept of gateways to show routing of data in a business process. A proposal was made to extend the BPMN standard to be usable for manufacturing domain in [10] by introducing four new types of gateways for depicting flows of materials (see Figure 13). This idea was used to model Material Gateways in the Process Insight Repository with some alterations.

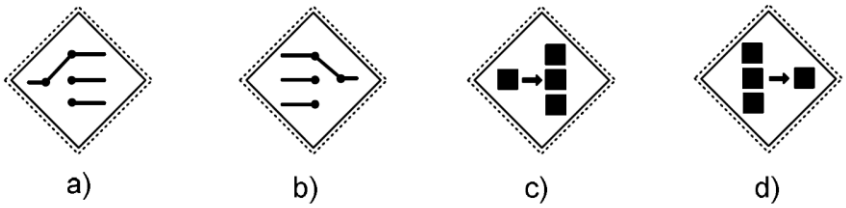

Figure 13: Material Gateways: Material Route Gateway (a), Material Select Gateway (b), Material Split Gateway (c), Material Join Gateway (d) [10]

Material Route Gateway represents a diversion point in the material flow, i.e. when one Production Step can be followed by one of the alternative Production Steps. Material Route Gateway models routing of material based on a decision and has one preceding Production Step and multiple following Production Steps (see Figure 13 a). [10]

Material Select Gateway represents selection of one of the incoming materials, i.e. when a Production Step is initiated by receiving materials from one of the preceding Production Steps. Material Select Gateway has multiple preceding Production Steps and one following Production Step (see Figure 13 b). [10]

Material Split Gateway creates parallel flows by dividing material and directing it to multiple following Production Steps. Material Split Gateway has one preceding Production Step and multiple following Production Steps which are executed in parallel (see Figure 13 c). [10]

Material Join Gateway represents joining of incoming flows by waiting for the completion of all preceding Production Steps before initiating the following Production Step. Material Join Gateway has multiple preceding Production Steps and one following Production Step (see Figure 13 d). [10]

In the conceptual schema of the Process Insight Repository the four described types of gateways are abstracted by a generic Material Gateway (see Figure 14). A generic Material Gateway can have multiple preceding Production Steps and multiple following Production Steps. A Production Step has zero or one input and output gateways, e.g. starting step of a Manufacturing Process would have zero input gateways and ending step would have zero output gateways. Material Gateway Instance represents the Material Gateway that was actually taken during the execution of a Manufacturing Process Instance.

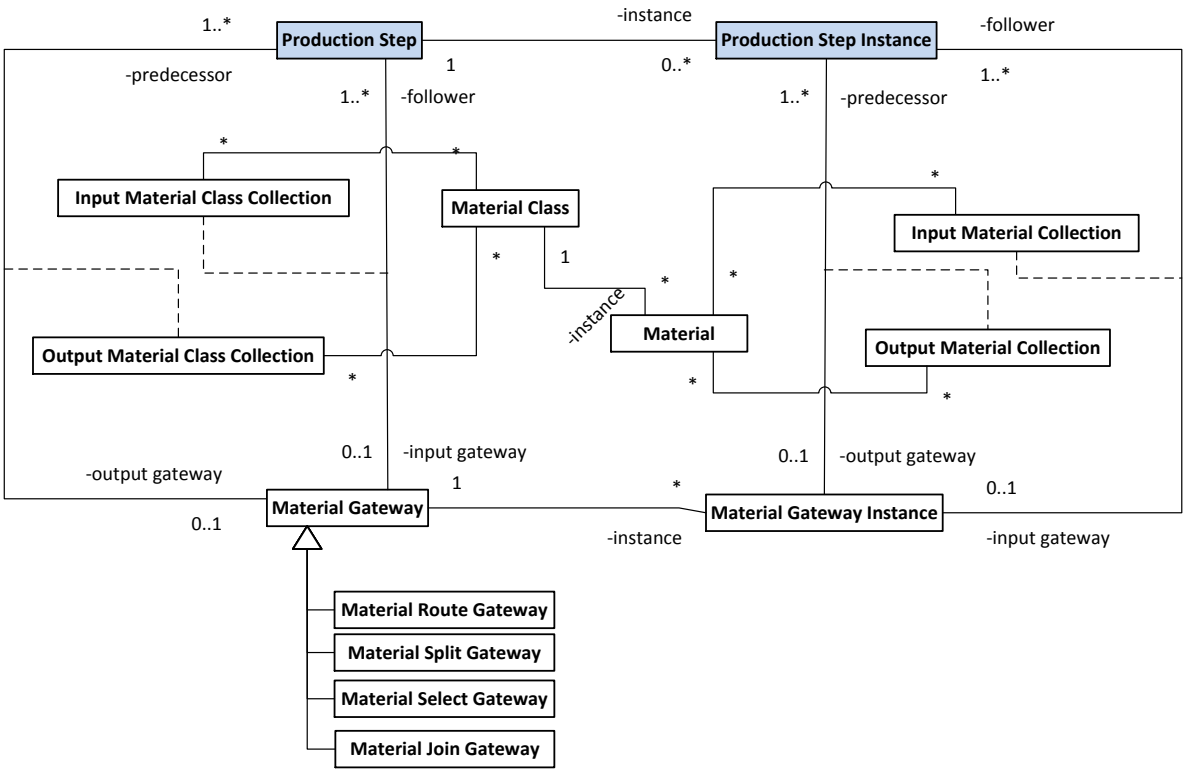

Figure 14: Material Gateways: control flow and material flow

Entities Material Class and Material represent types and instances of materials that can be transferred between Production Steps. The relation between a Material Gateway and a Production Step has associative entities Input Material Class Collection and Output Material Class Collection, which represent a collection of materials that are transferred from or to a Production Step. Associative entities Input Material Collection and Output Material Collection play the same role but in the run-time model of the process.

This model of Material Gateways allows storing the detailed information about materials that are transferred between Production Steps and allows modeling of sequential, parallel and conditional execution of Production Steps. Material Gateway Instance can also hold the information on which a decision of a Material Gateway was made, e.g. “Production Step X was initiated because the result of quality test showed defects”. This model of Material Gateways can also be more representative to the user, i.e. it can be mapped to a graphical notation, such as proposed in [10], and rendered to the user in an easy to understand graphical manner.

2.2. Insights

The previous Section 2.1 discussed the conceptual schema of a manufacturing process model. This Section is dedicated to the conceptual schema of insights into manufacturing processes. In terms of this work, an insight is knowledge about some aspects of a manufacturing process. Insights are stored in a formalized way in the Process Insight Repository in order to be available for the user.

2.2.1. Kinds of Insights

An insight can represent various kinds of data. Several kinds of insights are defined for the Process Insight Repository in terms of this work: Metric, Free Form Knowledge, Data Mining Model and Special Construct (see Figure 15). The following sections describe these kinds of insights in detail.

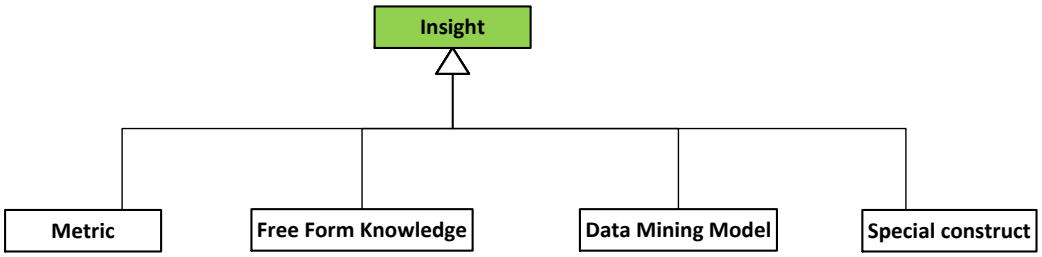

Figure 15: Kinds of Insights

Metric

A Metric is a measurement of a particular characteristic of a manufacturing process performance or efficiency. Examples of Metrics are production cycle time, rework percentage, product quality and others. [11] A Metric can be a numeric or nominal measurement and has a measurement unit that is specified by association with a Unit entity, e.g. cycle time is a real value measured in seconds and product quality can be a nominal value with predefined labels “GOOD” and “DEFECT”. A Metric is associated with the characterized object of the manufacturing process model with an Insight Association (see Section 2.2.2). Each Metric

belongs to one of Target Dimensions (see Figure 16), e.g. “time”, “economy” or “power consumption”. A Calculation Rule is associated with a Metric to specify the formula for calculating the value of the metric.

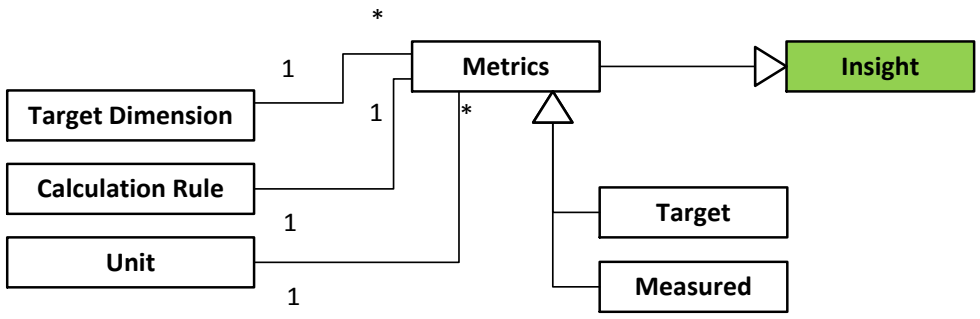

Figure 16: Metric

Metrics are classified to Measured and Target metrics. A Measured Metric is a value that was actually recorded during the execution of a Manufacturing Process, while a Target Metric is a value that should be achieved during the execution of a Manufacturing Process.

Target Metrics could be also extended with properties for automatic assessment of the manufacturing process performance by comparing values of Measured Metrics to Target Metrics. This could be done by specifying how the value should be used by the assessment algorithm, e.g. “maximize”, “minimize” or “not more than”. This approach however would narrow the usage of these target values, since estimation rules can be complex and can require aggregation of multiple Measured Metrics. For example, a Target Metric “production line power consumption <= 250 kWh per day” would require aggregation of power consumption by all machines that that were used for a particular Manufacturing Process. This Target Metric can be easily calculated using the data available in the Process Insight Repository; however, it is too complex to model all the possible expressions that can be used in calculation rules. Hence, it was decided that the target values would simply hold the value of the Target Metric and the automatic assessment would be performed using application logic.

Special Constructs

Special constructs exist on the scheme, but at the moment they cannot be applied to a manufacturing process. Special constructs, as mentioned in [5], are analogs of workflow patterns, such as decision, early knockout and others. These constructs exist in most of the workflows, but since a manufacturing process is usually a linear process of processing a part through passing it over a line of machines, these constructs lose their meaning. However, it was decided to leave special construct as a particular case of Insight on the scheme. A future work can find applications of special constructs to manufacturing process.

Free Form Knowledge

Free Form Knowledge represents any unformalized knowledge that can be stored in electronic form. Although Free Form Knowledge cannot be used by computer algorithms, it can be useful for knowledge exchange between users of the Process Insight Repository. The following types

of Free Form Knowledge were defined for the Process Insight Repository: Text, Photo, Video and Audio (see Figure 17). Examples of Free Form Knowledge can be a reference manual for a particular machine type, a photo taken on a factory or a textual description of some Production Steps.

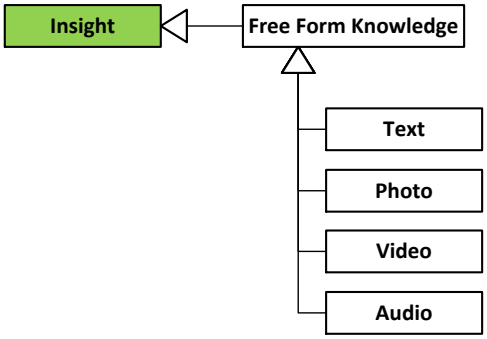

Figure 17: Free Form Knowledge

Data Mining Model

As was written in Section 1.2, the Process Insight Repository must provide support for performing data mining techniques on the manufacturing process data. Hence, the conceptual schema must contain entities that are related to data mining tasks. The main entity of a data mining task is the Data Mining Model. In order to provide support for main data mining techniques, the following types of Data Mining Models exist on the conceptual schema: Regression, Association, Clustering, Classification, Sequence analysis and Time series analysis (see Figure 18).

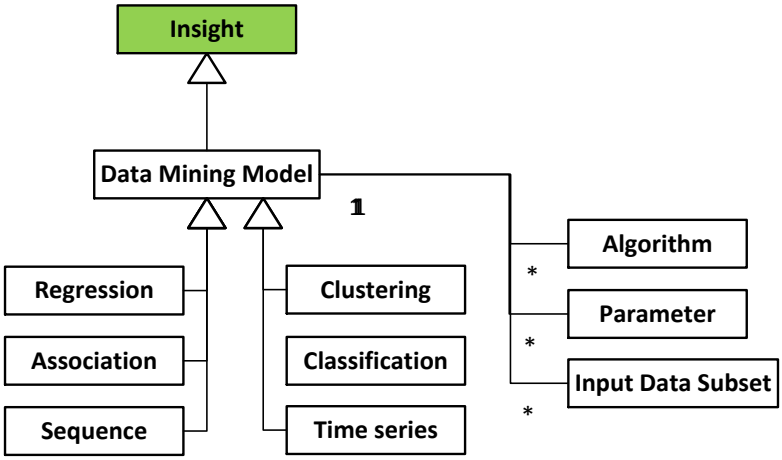

Figure 18: Data Mining Model

Any Data Mining Model that is stored in the Process Insight Repository is associated with an Algorithm entity, which describes the data mining algorithm that was used to derive the Data Mining Model (see Figure 18). For example, a decision tree classification model can be created using algorithm C4.5. Algorithm-specific parameters that were used in training of the Data Mining Model are stored in the associated Parameter entity, e.g. Boolean parameter “prune

decision tree" for C4.5 algorithm or integer number of clusters for k-means clustering algorithm.

The source data that was used for training a Data Mining Model is specified by Input Data Subset entity. It is important to specify the Input Data Subset for two reasons. The first reason is that the user must be able to find existing Data Mining Models for a given entity of the manufacturing process model, e.g. find Data Mining Models that are relevant for a particular Manufacturing Process Instance. The second reason is that the user must be able to find the source data on which the Data Mining Model was trained.

There are several possibilities to define the Input Data Subset:

- A predicate filter, e.g. "process_id=53 AND process_instance_timestamp between (11.09.2012 08:00 AND 11.09.2012 11:00)".
- Association to the input data, e.g. using a foreign key association in a relational database.
- Copy of the input data.
- Informal text description, e.g. "this model is trained on the data from executions of process X in period 11.09.2012 08:00 – 11:00"

A predicate filter and association with the input data can be used automatically and work both for finding Data Mining Models given an entity of the manufacturing process model and for finding the source data of the Data Mining Model. Storing a copy of the input data with the Data Mining Model is generally not efficient, because in this case the connection to the original source data is lost and in case of big volumes of accumulated process data this would lead to inefficient consumption of storage space. Informal description in plain text can be used only as information for the user about the source data of a Data Mining Model. Hence, in order for the Input Data Subset to be usable automatically it is recommended to specify the Input Data Subset either using predicate filters or using associations to the input data.

2.2.2. Insight associations

In general, any entity of a manufacturing process model described in Section 2.1 can have associated insights. However, the Process Insight Repository can be viewed as an event-centric platform as most of the data that is accumulated about manufacturing processes belongs to Production Steps and Manufacturing Processes. Therefore it was decided to have insight associations only for the following entities of the process model: Manufacturing Process, Production Step and Operating Resource. Insight associations were defined for these entities as subclasses of the Insight entity (see Figure 19). The purpose of these subclasses is to show which kinds of insights can be associated with which entities, because not all kinds of insights can be applied to some classes of the Process Insight Repository. For example, Target Metrics can be defined for a Manufacturing Process, but not for a Manufacturing Process Instance, while Measured Metrics can be defined only for a Manufacturing Process Instance, but not for a Manufacturing Process.

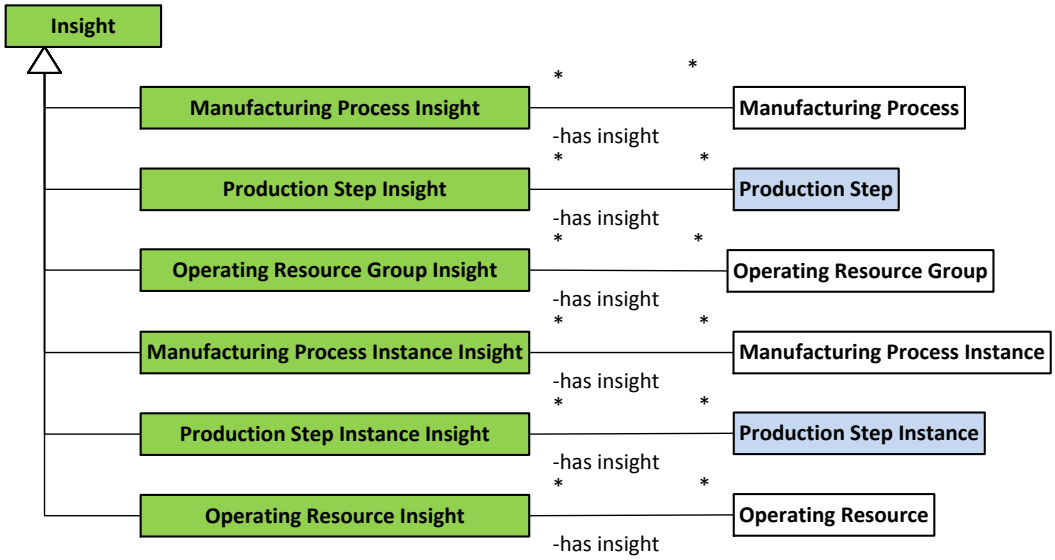

Figure 19: Insight associations

Any Insight that is stored in the repository can be associated with multiple objects of the manufacturing process model, e.g. a measured metrics “parts produced” can be associated with the corresponding product and the product group at the same time, a target metrics “availability” can be associated with a production line, a work unit and a particular operating resource.

Target Metrics define metrics that should be achieved when executing manufacturing processes. Therefore it makes sense to define them only for the entities of the build-time model: Manufacturing Process, Process Step and Operating Resource Group.

Measured Metrics appear when executing a manufacturing process. Initially they are associated with the entities of the run-time model; these are Manufacturing Process Instance, Production Step Instance and Operating Resource. These Measured Metrics represent direct measurements that were taken during the execution of a manufacturing process.

Measurements of the run-time model can be aggregated over a set of run-time model entities (e.g. Process Instances of a particular Process that were executed within a specified period of time). Although the aggregated measurements are taken from the run-time model, they characterize the corresponding entities of the build-time model. To store these aggregated measurements, Measured Metrics can also refer to the entities of the build-time model.

Free form knowledge may be associated with any object of the process model. However, it is not practical to store knowledge for a Process Instance or Process Step Instance, because that knowledge will refer only to a particular execution of the process and probably will not be reused. However, even when that knowledge was obtained from a particular Process Instance, it is better to associate it with the Process, but not with the Process Instance. In this case it will be easier to find this knowledge and use it.

Data Mining Models are generated from data of the real-time process model, but the resulting data mining models, similarly to Free Form Knowledge, are associated with the entities of the build-time model.

The summary of the described associations is shown in Table 1 (symbol x means existing association).

Table 1 Association of Insights with entities of the manufacturing process model

		Target Metrics	Measured Metrics	Free form knowledge	Data Mining Model
Build-time model	Manufacturing Process	x	x	x	x
	Production Step	x	x	x	x
	Operating Resource Group	x	x	x	x
Run-time model	Manufacturing Process Instance		x		as source data
	Production Step Instance		x		as source data
	Operating Resource		x	x	

On the conceptual schema of the Process Insight Repository these associations are shown using different kinds of Insights (see Figure 20), e.g. Manufacturing Process has association with Manufacturing Process Insight, Manufacturing Process Instance with Manufacturing Process Instance Insight. It helps to better understand which kinds of insights are stored for which objects. However, for the implementation simplicity all these Insight subclasses can be represented with one generic Insight class that would be associated with the entities of the manufacturing process model.

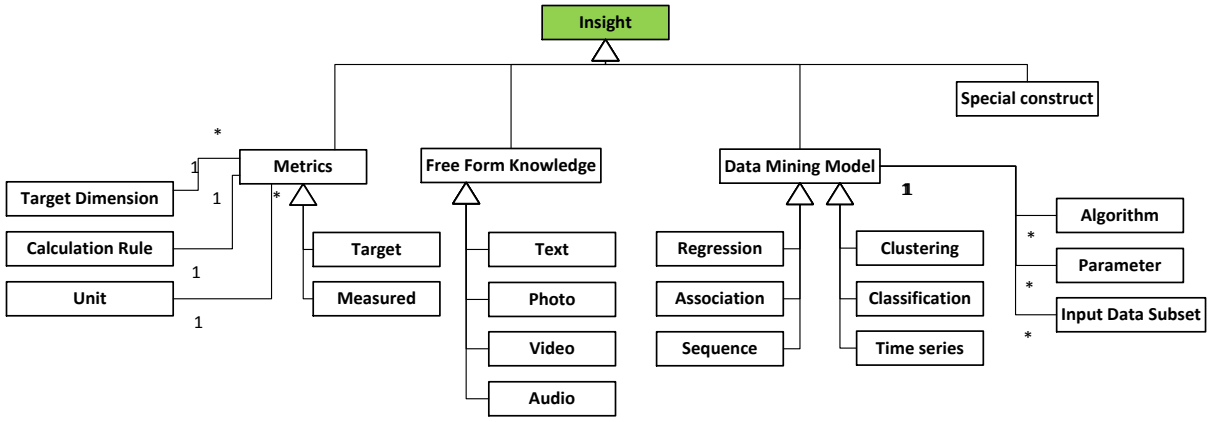

Figure 20: Summary kinds of Insights

2.3. Complete conceptual schema of the Process Insight Repository

The complete conceptual schema of the Process Insight Repository is given in Figure 21. The schema of the manufacturing process model presented in this work is designed in a normal form. This has a benefit of allowing preparing data for different kinds of analysis depending on the needs. The schema of the manufacturing process model can be viewed as a snowflake schema. Depending on the type of analysis, different entities can be taken for facts. For example, if a Process Instance is taken as a fact, then Production Step Instances, Machines and Employees used in the Process Instance can be viewed as dimensions of a Process Instance.

The designed conceptual schema is not final and can be extended depending on the available manufacturing process data. Attributes of the manufacturing process are not modeled in the conceptual schema, because they depend on the manufacturing process data that is available. Hence, the Process Insight Repository must be implemented in a way that allows changes in the model of a manufacturing process. The details of such implementation are discussed in Chapter 4. The technologies that can be used for storing different parts of the conceptual schema are discussed in Chapter 3.

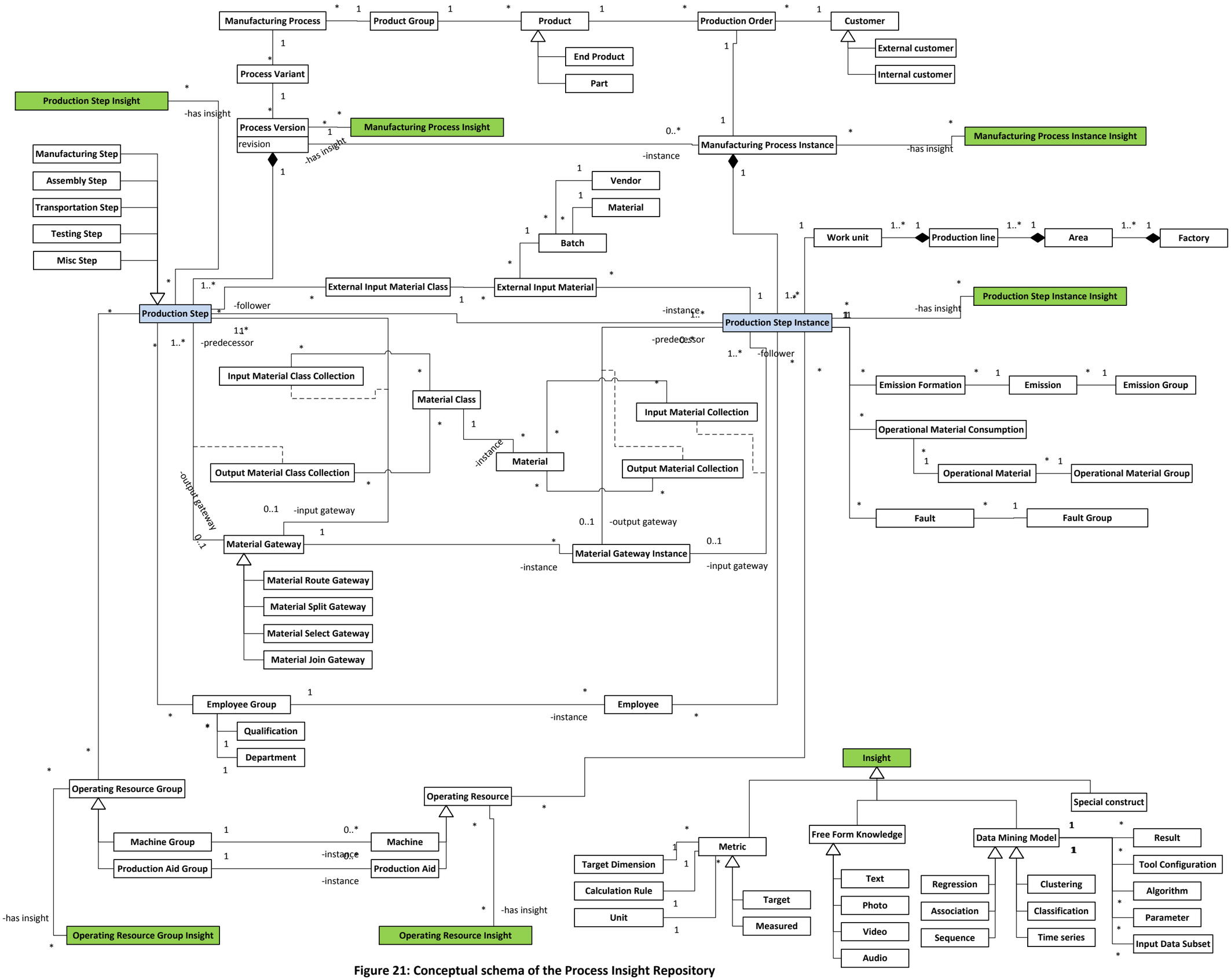

Figure 21: Conceptual schema of the Process Insight Repository

3 Technologies to implement the Process Insight Repository

The Process Insight Repository stores several different kinds of data. This introduces a challenge to find a proper way to store all the data using appropriate technologies. The data should be stored using the most suitable technology and must be accessible from the repository and connected to other parts of the repository. An observation of technologies for storing the Process Insight Repository is made in this chapter. The whole repository can be roughly divided into three big parts with regard to the type of the data.

The first kind of the data is structured data of the manufacturing process model. This is the part of the Process Insight Repository that stores the process model, which consists of the build-time and run-time process models. A part of the Measured Metrics can be stored directly in the process model as well (e.g. execution time of a Production Step Instance can be stored along with the object itself). One characteristic of this data is that it has a clear structure. Another characteristic is that this data can be large in volume. This results from the fact that the run-time process model stores the history of all the process executions on the factory. The schema of the manufacturing process model is flexible and can change. For example, new entities can be added to the schema or existing entities can be extended with new attributes. In order to handle these changes of the manufacturing process model, its schema must be described by some metadata. Manufacturing process model metadata must formally describe entities of the process model, the relations between them and properties of attributes.

The second part of the data is Data Mining Models and all the structures that are associated with data mining tasks performed by the Process Insight Repository. The technologies for this part of the data must be properly selected in order to allow exchange of data mining models and integration of external data mining tools into the Process Insight Repository.

The third kind of the data is the Free Form Knowledge part of Insights. The features of this kind of data are that it is not structured and can include multimedia content. This data will not be used automatically, but only by the user for the purposes of knowledge exchange. Hence, the user must be able to find the required Free Form Knowledge and retrieve it from the Process Insight Repository.

3.1. Storing manufacturing process model data

As was shown in Section 2.1, the data of the manufacturing process model is well structured, but the structure is not fixed and can change with time, e.g. when new entities or attributes are added to the model. In general, the data of the manufacturing process model can be stored using any solution suitable for storing structured data, such as relational database systems or XML files.

The main use of the manufacturing process data in the Process Insight Repository is to be available for the user and to be prepared for the data mining tool. In order to be usable for the data mining tool, the data must be either stored in a form that can be accessed by the tool or it must be transformed to a form that can be accessed by the tool.

In order to select technologies for storing the manufacturing process model in the PIR, several existing manufacturing and business process modeling tools were analyzed in order to find out the ways this problem is solved in these products. The overview of the three selected process modeling products is given in the following section. The questionnaire for process modeling tools contains the following questions:

- How do these tools represent and store process models internally?
- What means of exchanging process models do they have?

3.1.1. Overview of process modeling products

Process Designer

Process Designer is a CAD product of Siemens PLM Software for planning of assembly lines and processes. It is classically designed to use a central relational database system (Oracle Database) in order to provide a parallel multi-user environment. Each user of the system has a local working copy that is stored in a proprietary format and synchronized to the central database. Hence, all the process model data is stored in a central relational database. [12]

As means of exchanging process models, Process Designer supports export and import using an XML-based eBOP (electronic Bill of Process) format.

ProcessMaker

ProcessMaker is an open source business process management (BPM) or workflow software application. Workflow software such as ProcessMaker assists with designing, automating and deploying business processes or workflows of various kinds. Although ProcessMaker is not supposed to be used for modeling of manufacturing processes, business processes have in general common structure with manufacturing processes. [13]

ProcessMaker stores workflows in a database backend, which can be one of the supported relational database management systems, including MySQL, PostgreSQL, Oracle Database and SQL Server. [14] ProcessMaker uses an XML-based format for exchanging process models. [15]

Virtual Factory Framework

Virtual Factory Framework is a collaborative research project funded by the European Commission. One part of the project was development of a Virtual Factory Data Model, which is a data model based on Web Ontology Language and Resource Description Framework. The Virtual Factory Framework project designed a reference data model consisting of a set of ontologies that are based on IFC standard and describe several domains of manufacturing, including physical structure of the factory, products, manufacturing processes, production resources and others. The project proved the concept of using Semantic Web for modeling of manufacturing process data. [16] The project also has a set of prototype tools which work with a shared repository of ontologies. [17] Hence, such ontology-based data model can be used in the Process Insight Repository to store the build-time and run-time process models.

Storing build-time process model can bring the benefit of having good flexibility of the process model as using ontology allows describing knowledge in a flexible way. Run-time part of the

manufacturing process model contains big volumes of data, thus performance of a selected Web Ontology Language framework must be first analyzed in order to find out if the ontology framework can effectively handle big volumes of data. Storing process model as ontology also brings the overhead of data transformation, because ontology data cannot be directly accessed by a data mining tool and the PIR must prepare the manufacturing process data for data mining tasks.

Web Ontology Language operates on a set of "individuals" with a set of "property assertions" which relate these individuals to each other. These property assertions can later be used by semantic reasoner to infer logical consequences. [18] This might not give benefits for storing the process model, as it is a well-structured data. However, the facilities of the semantic reasoner can be helpful in organizing free form knowledge and searching for insights in the Process Insight Repository.

3.1.2. Summary on technologies for storing manufacturing process models

From the observed products and standards we can make a conclusion that the main used storages for managing process model data are technologies for storing of structured data. Object-relational database management systems are used for internal storage of data, while various XML-based formats are mostly used when process models must be accessed from outer systems (e.g., for exchange of process models between different systems). A storage technology based on semantic web can also be used. However, the benefit of using semantic storage for well-structured process data is questionable and can bring overheads.

In general, any storage of structured data would suffice for the needs of the Process Insight Repository. It is recommended to use a database management system, because it will provide easy access to data and good performance for big volumes of run-time part of the manufacturing process model of the Process Insight Repository. For means of exchanging process models, the Process Insight Repository can implement export and import in one of XML-based formats. For example, this can be used for initial setup of a manufacturing process in the Process Insight Repository if the process model already exists in a process modeling tool.

3.2. Storing data mining models

3.2.1. Data that needs to be managed in data mining processes

As discussed in introduction, in order for the PIR to implement data mining use cases, it must encapsulate data mining tasks and provide a simple API to applications. The user should not specify any parameters, which are specific to data mining, thus all parameters and settings of the data mining process must be stored in the PIR.

In order to define what data mining related data needs to be stored, let us first see which tasks must be performed by the PIR to implement the data mining use cases described in the introduction. There exist several data mining methodologies that define stages of the knowledge discovery process. According to the results of the poll on data mining methodologies [19], the most popular standardized methodologies for knowledge discovery are: CRISP-DM (42%), SEMMA (13%) and the KDD Process (7%).

KDD (Knowledge Discovery in Databases) is a process of using data mining methods to extract knowledge using a database along with any required preprocessing, sub sampling and transformation of the database. [20] SEMMA (Sample, Explore, Modify, Model and Assess) is a knowledge discovery methodology introduced by SAS [21]; it contains some of the essential elements of any data mining project, however, it is aimed for users of SAS Enterprise Miner software and concerns only the statistical, the modeling, and the data manipulation parts of the data mining process. [22] CRISP-DM (Cross Industry Standard Process for Data Mining) is a data mining methodology developed by a consortium of data mining vendors and companies. [23]

The three methodologies mentioned above are similar in general. Table 2 shows a summary of correspondences between CRISP-DM, SEMMA and KDD [24].

KDD	***SEMMA***	***CRISP-DM***
- (Pre KDD)	-	Business understanding
Selection	Sample	Data Understanding
Pre processing	Explore	
Transformation	Modify	Data preparation
Data mining	Model	Modeling
Interpretation/Evaluation	Assessment	Evaluation
- (Post KDD)	-	Deployment

Table 2: Correspondence of data mining stages in different data mining methodologies [24]

CRISP-DM, besides being the most popular methodology, also provides the most comprehensive approach to the knowledge discovery process [24]. In particular, it covers Deployment phase, which is not covered by KDD and SEMMA. For these reasons CRISP-DM is taken as a reference methodology for performing data mining tasks in the PIR. CRISP-DM defines the following six stages of knowledge discovery process: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment [23] [25].

As mentioned above, the Process Insight Repository must store all the data that is necessary for data mining tasks and implement some stages of the data mining process. A more detailed overview of CRISP-DM data mining methodology is given below in order to shows which data is needed and where it can be stored for each stage of a data mining process.

The first two steps of CRISP-DM, Business Understanding and Data Understanding are performed by a data mining expert. The data that is needed for these two steps is the data contained in the run-time and build-time parts of the process model and some of insights (e.g., free form knowledge about existing processes and process metrics). This data is available in the PIR and can be accessed by the data mining expert using a user interface application via navigation API of the PIR (see Section 4.4).

A data preparation step is necessary to transform the data contained in the PIR to a form that can be used by a selected data mining tool (see Section 4.5). The PIR must provide support for data transformation. This functionality will be used by the PIR itself and by the data mining expert when performing manual knowledge discovery processes. Parameters of transformation must be stored along with the Data Mining Model in the PIR. These parameters include the

specification of the source data, predicate filters to select particular source data, rules to pre-process the data, assigning roles to attributes and other parameters.

In the Modeling step a new data mining model is trained on the data which was prepared in the data preparation step. The main parameters for this step are the algorithm to create the data mining model and various algorithm-specific parameters supported by this algorithm. According to use cases described in Section 1.2, the algorithm and its parameters are not in competence of a non-expert user, thus they must be either predefined by the data mining expert or there should be some reasonable defaults. The advanced user is a more experienced user, thus the PIR should allow this type of user to alter some of the parameters of the data mining algorithm. The parameters of the data mining algorithm must be stored along with the created data mining model in order to know how the particular data mining model was generated. The output of this data mining phase is the derived data mining model.

The Deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process. [25] In the Deployment phase the generated data mining model must be stored in the PIR and made available for the users of the PIR. The PIR must provide functionality to apply stored data mining models to the new data on user request. In order to do this, the new data must be transformed the same way as in the data preparation step. Thereby all the data transformation parameters used in the data preparation step must be stored along with the data mining model. The results of applying the data mining model (e.g. assigned classification labels or predicted numeric values) should be either returned to the user or stored in the PIR.

The summarized list of all kinds of data that need to be stored in the PIR with regard to data mining use cases is: data transformation parameters; data mining algorithm; algorithm-specific parameters; data mining model; results of applying data mining model to the new data.

3.2.2. Standards for storing and exchanging data mining models

The literature review on standards for storing and exchanging data mining models showed that there are two main standards for exchanging data mining models. These standards are PMML standard defined by Data Mining Group [26] and CWM-DM standard defined by Object Management Group [27].

PMML

Predictive Model Markup Language (PMML) is an XML-based markup language that provides a vendor-independent way for applications to define models related to predictive analytics and data mining and to share those models between PMML-compliant applications [28]. PMML was developed by the Data Mining Group, a vendor-led committee composed of commercial and open source analytics companies, and is now considered to be the de facto standard language used to represent data mining models [28].

PMML allows one to easily share data mining models between different applications. A predictive data mining model can be trained in one system, expressed in PMML, and then moved to another system where it can be applied to new data [28]. PMML is supported by a

wide range of applications [29]. Hence, it can be used in the Process Insight Repository to store created data mining models in a format that can be used by multiple data mining tools.

A PMML data mining model incorporates data pre-processing, the predictive model itself and data post-processing (see Figure 22) [28]. PMML specifies the format of the input data, but does not specify where this data must be taken from. The same is true for the output of the PMML model application. Hence, if PMML is used for storing data mining models, the Process Insight Repository must store additional information along with the PMML data mining model, i.e. mapping of the manufacturing process model data to the PMML input, mapping of PMML output data and the source data on which the data mining model was trained. An example of implementing these mappings is shown in Section 4.3.

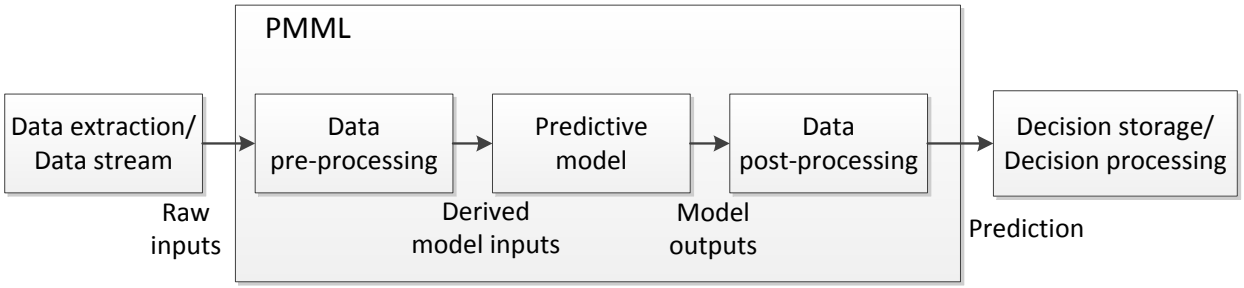

Figure 22: PMML representation of a data mining model [28]

CWM-DM

The Common Warehouse Metamodel (CWM) is a specification that describes metadata interchange among data warehousing, business intelligence and knowledge management technologies [30]. The Common Warehouse Metamodel consists of a number of sub-metamodels which represent common warehouse metadata in major areas of interest. One of the sub-metamodel is CWM Data Mining, which represents three conceptual areas of data mining: the data mining Model description itself, Settings and Attributes. The data mining Model description represents a mathematical model produced or generated by the execution of a data mining algorithm. The Settings conceptual area elaborates on the settings of a data mining algorithm. The Attributes conceptual area describes data mining attributes [30].

In spite of the fact that CWM-DM works out data mining models in more details compared to PMML, CWM-DM is rarely integrated in data mining solutions due to the complexity of its model. [31]

3.2.3. Selection of data mining products for the Process Insight Repository

In order to perform data mining tasks, the Process Insight Repository should use some data mining tools. This section selects a set of data mining products based on several criteria. Sections 3.2.4 and 3.2.5 provide an overview of the selected data mining products.

There exist many products to perform data mining tasks. In terms of this work, it is not necessary to provide a complete analysis of all available data mining tools. Hence, we applied a two-stage selection process. The tools that were selected in this first step were analyzed in more detail to verify if they can be used with the Process Insight Repository. An example

methodology for the selection of data mining tools can be seen in [32]. This methodology provides a comprehensive set of questions that need to be answered to select a proper data mining tool. In order to make the first selection, the following two factors were considered:

- Is the product provided as a part of a database system or is it a third-party tool with regard to the database system? Products from both categories are to be selected for verification.
- Is the product proprietary software or is it open-source software? Products from both categories are to be selected for verification.

Thereby, the most popular products from these two categories were selected on the first step based on the poll results on the popularity of data mining tools [33]. These data mining products are presented in Table 3.

Table 3: First selection of data mining products

	Proprietary product	***Open-Source product***
Part of the database system	Oracle Data Mining (Oracle Database 11g) IBM InfoSphere Warehouse Microsoft SQL Server 2008 Analysis Services	PostgreSQL MySQL
Not part of the database system	Excel SAS Enterprise Miner IBM SPSS Modeler	RapidMiner KNIME Weka

A more detailed overview of the selected data mining products is presented in the two following sections in order to find out whether these products fit the need of the Process Insight Repository and how they can be used in it.

A data mining tool must support all the data mining techniques that need to be performed in the PIR (see Section 2.2.1). If it is not possible to cover all the data mining techniques with a single tool, then a set of data mining tools can be used to provide complete support of data mining in the Process Insight Repository.

3.2.4. Overview of data mining support in database systems

Performing data mining tasks directly in the database can provide better performance and is therefore very attractive technology. The reason of performance benefits is that the data, data mining models and results are stored and processed in the database, hence eliminating data movement and minimizing information latency. To have a uniform overview of data mining support provided by different database systems, a set of questions was answered for the selected database systems. The questions are:

- Which support for data mining is provided?
- How does it differ from SQL/MM standard?
- Which data mining techniques and algorithms are supported?
- How are data mining models and results stored?

- Which export and import formats are supported for exchanging data mining models?
- How can it be used in the Process Insight Repository?

ISO/IEC 13249-6: SQL Multimedia and Application Packages (SQL/MM)

SQL Multimedia and Application Packages (SQL/MM) is a standard that complements the Structured Query Language (SQL) defined by ISO/IEC 9075 standard. SQL/MM defines SQL based interfaces and packages to support multimedia data, such as full-text data, spatial data, and still images. The SQL/MM Part 6 addresses data mining and provides an API for data mining applications to access data from SQL/MM compliant relational databases. [34] [35]

The standard supports four different data mining techniques: Rule model, Clustering model, Regression model and Classification model. Every model has a corresponding SQL structured user-defined type. A set of routines is defined to manipulate these user-defined types. These routines allow to set parameters for data mining activities, training of data mining models, testing of data mining models and application of data mining models. ISO/IEC 13249-6:2006 does not specify the import, export or exchange of data mining models between different systems. However, it references the Predictive Model Markup Language specification. [30] [35]

Oracle Data Mining (Oracle Database 11g, 10g)

Oracle Data Mining (ODM) is part of the Oracle Database product. Oracle Data Mining provides data mining functionality as native SQL functions within the Oracle Database. It allows building and applying predictive data mining models using one of the provided interfaces, which include Java programming interface (Java Data Mining), PL/SQL interface and Data mining SQL functions [36]. Implementation of Data mining SQL in Oracle Database does not follow the SQL/MM standard described above. Oracle Data Mining supports main predictive and descriptive data mining techniques with a number of data mining algorithms. Data mining models and parameters are stored internally as relational structures and PL/SQL packages in the database.

Oracle Data Mining can export and import data mining models. However, models can be exchanged mainly in a proprietary format, while support of PMML format is very limited. Only some particular types of data mining models can be exported and imported in PMML format. Namely, only decision tree models can be exported as PMML v2.1. Oracle Data Mining supports the core features of PMML 3.1 for import of regression models. Import of PMML models is supported only for regression models for either linear regression or binary regression. Oracle recommends using export and import to dump files (in a proprietary format) for deployment and migration of data mining modes. [37]

IBM InfoSphere Warehouse

IBM InfoSphere Warehouse is part of IBM DB2 version 10, a database management system developed by IBM Corporation. This product can be formerly known as IBM DB2 Data Warehouse Edition (in DB2 version 9) and IBM Intelligent Miner (in DB2 version 8). IBM InfoSphere Warehouse supports main data mining techniques. The data mining functionality can be used via SQL programming interface, which conforms to SQL/MM standard. [38] No

literature sources were found about the way data mining models are stored internally in the database.

IBM InfoSphere Warehouse provides rich support for exporting and importing data mining models in PMML format. This means that different data mining tools can be integrated with the InfoSphere Warehouse using PMML as interchange format. For example, data mining models generated with an external data mining tool can be deployed to InfoSphere warehouse and applied to the new data directly in the database, thereby improving performance by eliminating data movement.

Microsoft SQL Server 2008 Analysis Services

Microsoft SQL Server Analysis Services (SSAS) is a tool for online analytical processing, data mining and reporting. It supports performing a range of data mining techniques in Microsoft SQL Server. It also supports accessing other sources of data, such as IBM DB2, Oracle Database, Microsoft Access and others.

Microsoft SQL Analysis Services enables data mining facilities by providing several interfaces. Data Mining Extension (DMX) is a query language extension for data mining models. It has SQL-like syntax, whereas SQL syntax operates on relational tables, DMX operates on data mining models. DMX is used to create and train data mining models and to browse, manage and predict against them. Microsoft SQL Analysis Services provides a .Net programming interface as well. [39]

Microsoft Analysis Services provides means to export and import data mining models and structures. However, this support uses a proprietary format and can be used only across different SQL Server Analysis Services instances. The support of PMML standard is limited to PMML version 2.1 and only Microsoft built-in algorithms are supported for decision tree and clustering models [40] [41]. According to reference manuals, export and import of models in PMML format is provided mainly as means of migrating data mining models from SQL Server 2000 Analysis Services to SQL Server 2008 Analysis Services.

PostgreSQL & MySQL

PostgreSQL is an object-relational database management system available for many platforms. It is developed by the PostgreSQL Global Development Group and released under PostgreSQL License, which makes it free and open source software.

MySQL is an open source relational database management system. It is released under GNU General Public License, as well as under Commercial License. At the moment of writing MySQL is owned by Oracle Corporation.

PostgreSQL and MySQL are the most popular open source relational database management systems. Literature review showed that PostgreSQL and MySQL do not yet provide facilities to perform data mining tasks directly in the database. However, majority of data mining tools support these database management systems as data sources. Hence, such open source database management systems can be still used for the implementation of the Process Insight

Repository as long as all the data mining functionality is implemented with external data mining tools.

Summary of data mining support in database systems

The observed database systems provide limited support of data mining techniques. Only commercial databases provide support for in-database data mining. The observed open source databases do not provide any support for data mining besides serving as storages of the source data for external data mining tools. Support of sequence analysis and time series analysis techniques is not provided by database systems.

From the observed data mining solutions from database vendors only IBM InfoSphere shows support of exporting and importing data mining models in a standardized PMML format. Other products have only limited support of PMML models which is restricted only to some kinds of data mining techniques. Existing means of export and import in proprietary formats are provided mainly for migrating data mining models and structures between similar systems and are not meant as means of exchange. All of the observed products do not support exchange of data mining models in CWM-DM format. The overview of data mining facilities provided by database systems is summarized in Table 4.

Table 4: Summary of data mining support by database systems

	SQL/MM Part 6	Oracle Data Mining (Database 11g)	IBM InfoSphere (DB2)	MS SQL Server 2008 Analysis Services	MySQL	PostgreSQL
Data mining techniques:						
Regression	1	1	1	1	no support	no support
Association	1	1	1	1		
Sequence			1	1		
Clustering	1	1	1	1		
Classification	1	1	1	1		
Time series				1		
PMML:						
PMML export	no	poor	good	poor	no support	no support
PMML import	no	poor	good	poor		

The in-database support for data mining techniques could be used for the implementation of the Process Insight Repository. In this case, data mining models created in the Process Insight Repository can be stored in the proprietary format of the database. However, this would limit the usage of these data mining models only to the database they were created in. In order to

store data mining models in a format that is usable for other systems, they can be stored in PMML format. However, only IBM InfoSphere provides good support of PMML, while other database systems provide only poor support of PMML. Using the in-database data mining in the Process Insight Repository brings the benefit of better performance when creating and applying data mining models, while data transfers are minimized. A conceptual schema of such integration is shown in Figure 23.

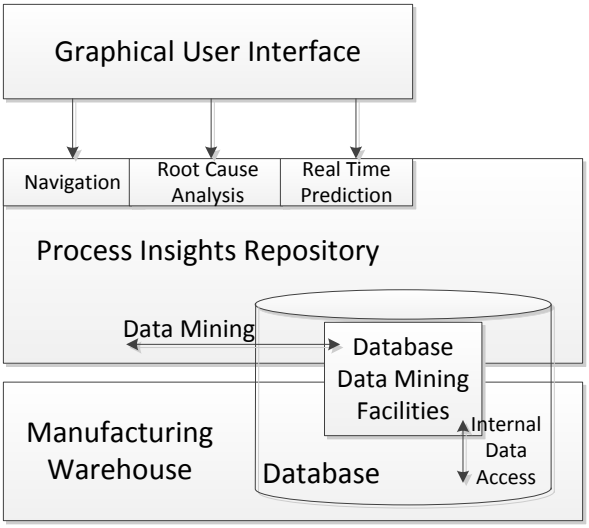

Figure 23: In-database data mining in the Process Insight Repository

3.2.5. Overview of data mining products

There exist a number of data mining tools that provide much richer support for data mining techniques compared with database management systems. Data mining tools provide the data mining functionality, but do not provide facilities for storing data. Hence, they can be used only in combination with the main storage of the Process Insight Repository and must be somehow integrated into it. This section provides an overview of the data mining tools that were selected in Section 3.2.3 considering the possibility of their integration into the Process Insight Repository. As far as these products provide better support for data mining and do not have own storage, a different set of questions was put for each of the products:

- Which data mining techniques does a product support?
- Which data sources can it access?
- What facilities provided are provided for import and export of data mining models?
- Does it provide an API for integration of the product into the Process Insight Repository?

IBM SPSS Modeler

IBM SPSS Modeler is a commercial data mining software application used to build predictive models. It was originally developed by SPSS Inc. and is now owned by IBM. IBM SPSS Modeler provides a visual interface for users and an application programming interface for embedding SPSS functionality into other applications [42]. SPSS Modeler can access data from multiple data sources, i.e. database systems, spreadsheets and flat files [43].

SPSS Modeler supports a wide range of data mining techniques and algorithms, including sequence data mining and time-series forecasting techniques [43]. Hence, SPSS Modeler covers the data mining techniques that are needed for the Process Insight Repository. SPSS Modeler also provides support for in-database data mining algorithms for IBM InfoSphere, Microsoft SQL Server and Oracle Database. This means that if the Process Insight Repository will use SPSS Modeler as a data mining tool and one of the mentioned database systems as storage for manufacturing process data, the API of the SPSS Modeler can be used to perform both data mining in SPSS and in-database data mining.

SPSS Modeler provides good support for export and import of data mining models in PMML format [44]. Database native data mining models can be exported for IBM InfoSphere Warehouse models, thus allowing creating data mining models in SPSS Modeler and deploying them to the database system for direct in-database application on new data.

SAS Enterprise Miner

SAS (originally Statistical Analysis System) is an integrated system of software products for business analytics provided by SAS Institute Inc. Data mining functionality is mainly provided with SAS Enterprise Miner, a software product that supports a wide range of descriptive and predictive data mining techniques, including sequence analysis and time series data mining [45]. Thanks to SAS/ACCESS software modules, data can be accessed from multiple various data sources, i.e. relational database management systems, Hadoop Distributed File System (HDFS), legacy systems and file-based data [46].

SAS Enterprise Miner provides a Java API for embedding data mining algorithms into applications. Data mining models created in SAS Enterprise Miner can be exported and imported in PMML format. Hence, SAS Enterprise Miner can be integrated into the Process Insight Repository using the provided API and data mining models can be stored in PMML format.

RapidMiner

RapidMiner is an open source system for data mining. RapidMiner is available as a standalone application for data analysis and as a data mining engine for the integration into other products [47]. RapidMiner provides support for a wide range of machine learning algorithms [48]. RapidMiner can be integrated into other applications as a library using the provided Java API.

RapidMiner has a set of input and output operators that allow accessing various data sources, including database systems, spreadsheets and text files [48]. As means of export and import, RapidMiner allows exporting and importing data mining models in PMML format and in RapidMiner XML-based format, as well as the complete data mining processes using XML-based format.

Weka

Weka is a collection of machine learning algorithms developed at University of Waikato, New Zealand. Weka is distributed under GNU General Public License and is therefore free and open source software. It is written in Java and can be easily integrated into any Java application.

Weka supports a wide range of data mining algorithms, including algorithms for sequence analysis and time series data mining [49]. Weka is a popular library providing data mining algorithms and is therefore available as an extension in other data mining products, such as RapidMiner, R and KNIME. Since Weka is used as a Java library, it can access any data sources that are available for a Java application.

Weka supports only import of data mining models in PMML format, but does not support export of PMML data mining models [29], [50].

KNIME

KNIME is an open source data analytics platform. It was initially developed by a team of software engineers at Konstanz University as a proprietary product for pharmaceutical industry. KNIME has a data pipelining concept similar to RapidMiner. KNIME has a graphical user interface that allows data preprocessing, modeling and data analysis and visualization. KNIME supports various data mining techniques with a number of algorithms [51]. Sequence analysis and time series analysis are available with external plugins. KNIME can access data from almost all JDBC/ODBC-compliant databases, text files and XML files [52]. Functionality of KNIME can be used with the provided Java API [53].

As means of data mining models exchange, KNIME supports import and export of data mining models in PMML format [54]. Most of the data mining modules in KNIME natively support PMML [55].

Summary on integration possibilities for data mining products

In general, all of the observed data mining products provide sufficient support of data mining techniques for the needs of the Process Insight Repository. All of the products provide approximately the same support for accessing various data sources, i.e. all of them can access popular database management systems, text files or receive data with the provided API. The summary of data mining support is presented in Table 5. Any of the observed data mining products can be integrated into the Process Insight Repository using the API that they provide. A conceptual schema of such integration is shown in Figure 24.

All the observed data mining products, except for Weka, provide good support of exporting and importing data mining models in PMML format. Hence, PMML can be used as a generic format for storing data mining models in the Process Insight Repository.

Table 5: Summary for data mining products

	IBM SPSS Modeler	SAS Enterprise Miner	RapidMiner	Weka	KNIME
Support of data mining techniques:					
Regression	yes	yes	yes	yes	yes
Association	yes	yes	yes	yes	yes
Classification	yes	yes	yes	yes	yes
Clustering	yes	yes	yes	yes	yes
Sequence	yes	yes	yes	yes	yes
Time series	yes	yes	yes	yes	yes
Support of PMML:					
PMML export	yes	yes	yes	no	yes
PMML import	yes	yes	yes	yes	yes

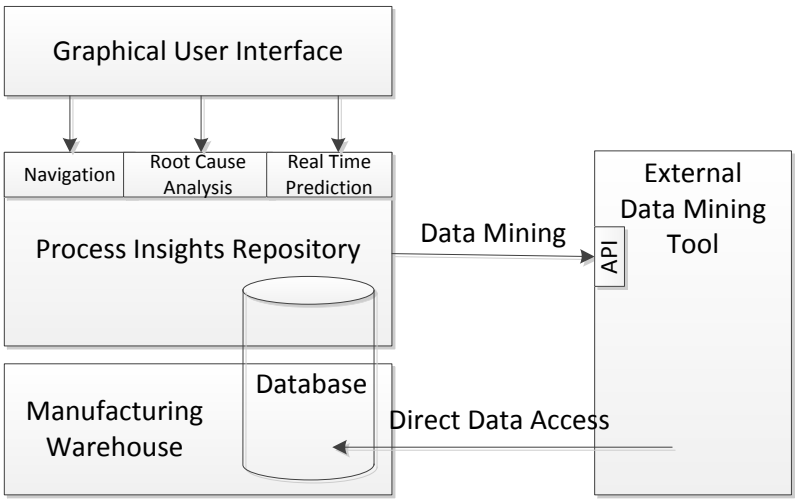

Figure 24: Integration of a data mining product into the Process Insight Repository

3.2.6. Summary on integration of data mining products

The overview of data base systems and data mining products showed that the only generic format for storing data mining models is PMML. Data mining support provided by databases can be used for the implementation of the Process Insight Repository. However, since data mining facilities provided by databases store data mining models in internal proprietary formats and provide only limited support for storing data mining models in PMML format, the data mining model created by a database cannot be exported and used by other data mining tools. Hence, the Process Insight Repository should use PMML for storing data mining models and

one of the data mining products that supports exporting and importing data mining models in PMML format. These products were listed in Sections 3.2.4 and 3.2.5.

All the observed products provide an API that is sufficient for programmatic integration of a data mining product in the Process Insight Repository, hereby eliminating user interaction with the data mining tool. Complete automation of the data mining process is important for the Process Insight Repository, because one of its requirements is that the user should not have any specific knowledge about data mining.

The Process Insight Repository can use in-database data mining as shown in Figure 23 or external data mining product as shown in Figure 24. To achieve better results, a combination of these two approaches can be used thanks to using PMML format for storing data mining models. An external data mining tool can be used to train data mining models and in-database data mining for the application of the trained data mining models to the new data. Using external data mining tools would give richer support of data mining, while application of these data mining models with the in-database data mining support will improve performance by eliminating unnecessary data transfers. An example of such architecture is shown in Figure 25.

If all the data mining tasks would be performed solely by an external data mining tool, then the support of the data mining by the database is not needed and almost any database can be used. In this case, the only restriction for selecting a database is that it must be supported by the data mining tool that is selected for the implementation of the Process Insight Repository.

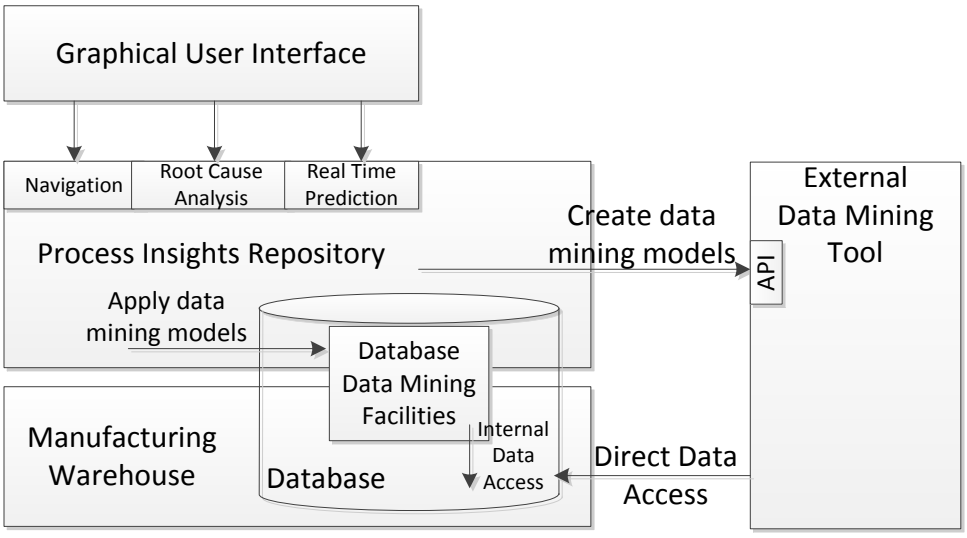

Figure 25: Combined approach – external data mining tool and in-database data mining

3.3. Storing free form knowledge

As discussed in Section 2.2.1, the Process Insight Repository must provide support for storing and accessing free form knowledge. Free form knowledge includes plain text, text documents in various formats, photo, audio and video documents. The functionality provided by the Process

Insight Repository must enable the client application to search and retrieve free form knowledge directly from the PIR, without the need to interact with any other systems.

The listed kinds of free form knowledge have different nature and representation to the user. However, they are treated jointly because they have some common properties. The common properties of free form knowledge are:

- Free form knowledge documents can have relatively big size, e.g. a text document in PDF format can take up to 10 MB and a video document can take up to hundreds of megabytes.
- Free form knowledge is usually represented as a document in a non-plaintext format, except for the plaintext documents.
- In order to be represented to the user, the complete document must be transferred as a binary file and a special application must be used, e.g. a PDF viewer for a text document in PDF format.
- Free form knowledge can be searched using the content, metadata or description of the document.

The main purposes of free form knowledge are to provide the knowledge exchange instrument for users and to give access to the existing free form knowledge. For example of knowledge exchange, a user can store the information that he discovered in a form of a text document or a photo that was taken on a factory and associate it with an object of the manufacturing process model. Example of access to the existing free form knowledge is giving access to instructions and manuals stored in PDF files.

In a simple case, an insight of free form knowledge would be associated with the objects of the process model that it is related to. However, it takes effort to associate big amounts of existing knowledge with the process model when introducing the Process Insight Repository into production. Examples of such documents can be operation manuals for machines that are used on the factory or various instructions. These documents might be left unassociated with the objects of the manufacturing process model, but need to be available and easily accessible for the end user. Hence, the Process Insight Repository must provide access to free form knowledge not only using insight associations, but providing search facilities as well.

3.3.1. Enterprise content management systems

One approach to handle free form knowledge in the Process Insight Repository is to store it in an Enterprise content management (ECM) system. An ECM system is a system designed to contain unstructured information such as files, images and drawings with the purpose to deliver the right content to the right person at the right time. Such systems typically consist of two components: an index repository which provides the search capabilities and a document repository where the actual documents are stored [56].

In this case all the documents would be stored in the ECM system, and the ECM system must be integrated to the Process Insight Repository using the provided API. For the search of documents by the user, the Process Insight Repository must wrap search calls to the ECM

system and provide a search API for the user. For the association of free form knowledge with the entities of the manufacturing process model, the Process Insight Repository must store an association with the identifier of a document in ECM system.

3.3.2. Storing documents in the database

Another approach to handle free form knowledge is to store documents directly in the database of the Process Insight Repository. In this case documents can be retrieved using full text search. Full text search is a technique of retrieving documents by performing search in the full contents of documents. Using this approach in the Process Insight Repository is possible due to the wide support of full text search facilities by database systems, i.e. all of the database systems observed in Section 3.2.4 provide support for full text search.

To enhance searching of documents, the meta-information of a document can be stored as plain text along with the binary document. Meta-information is a description of the document content in plaintext and can be used to find the document using a search query specified by the user. This meta-information can be either manually entered as text description when a user adds a document (e.g., entering description of a picture, taken on the factory) or automatically extracted from the binary document when it is possible (e.g., extract text of a PDF document). This plain-text meta-information is afterwards indexed by the full text search facilities of the database that is used in the PIR. It is not necessary to store meta-information for all types of documents, because some document types can be supported by the full text search of the database, e.g. popular document formats like Microsoft Office file formats or Adobe PDF family formats are supported by most of the full text search capable databases.

3.4. Summary on technologies for the implementation of the Process Insight Repository

3.4.1. Reference component schema of the Process Insight Repository

The summarized component schema of the Process Insight Repository is given in Figure 26. This reference component schema uses a relational database management system to store the data of the manufacturing process model. Hence, the data can be accessed by an external data mining tool directly. A combined approach is shown for the data mining tasks, i.e. data mining models are created with an external data mining tool and applied with the in-database data mining. The Free Form Knowledge is stored in an external Enterprise Content Management system.

3.4.2. Representing data to the user

In order to represent data mining models and data mining results to the user, this data must be processed (rendered) either by the user interface application or by the PIR. If representation of the model is done by the user interface application, data mining models and data mining results must be transferred in a standard form, for example using PMML standard. The user interface in this case must understand the generated PMML models and represent them to the user. The representation can be also done by the PIR if the data mining tool supports rendering of data mining models via API calls. In this case, the PIR must provide an API to render data mining models with the parameters required by user interface, and rendering must be done by the

data mining tool. This approach has a benefit that the implementation of the user interface is greatly simplified, because it does not have to perform representation of data mining models and data mining results. However, there is less control over rendering of models. An intermediate approach is also possible: the PIR can define data structures for supported data mining model types and data mining result types and returns results using these structures. The user interface must therefore understand these data structures and represent them to the user. The benefit of this approach is that different kinds of user interfaces can have different representation of data mining results, depending on the properties of user interfaces (e.g. depending on screen size and available input methods).

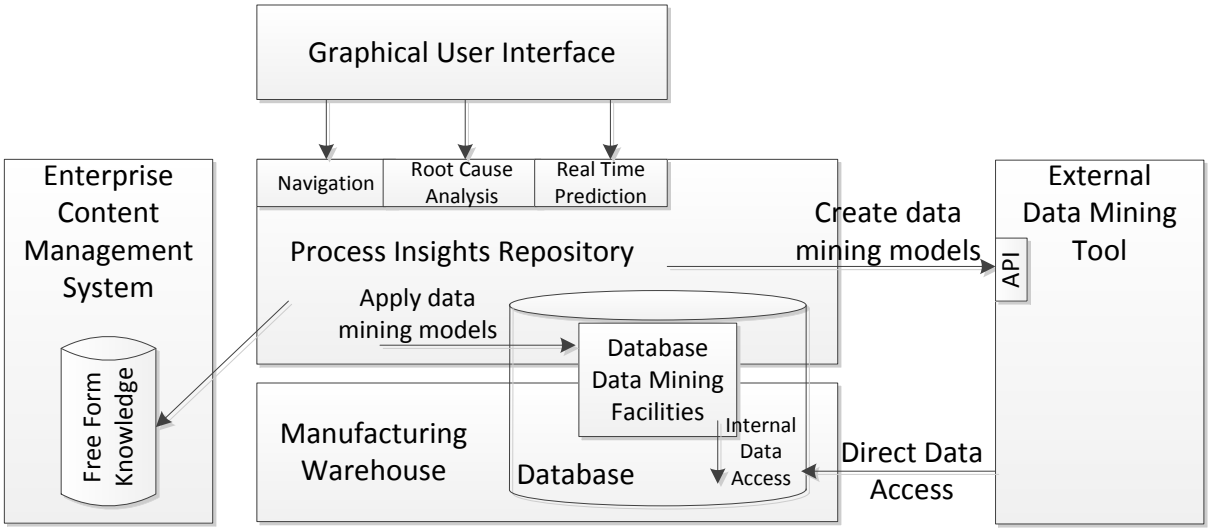

Figure 26: Reference component schema of the Process Insight Repository

4 Prototype Implementation

In terms of this work a prototype of the Process Insight Repository was implemented as a proof of concept. The schema of the prototype implementation follows the conceptual schema designed in Chapter 2 with some simplification (see Section 4.2). As far as there is no sample manufacturing data available, the prototype implementation has a module to generate random data of a sample manufacturing process described in Section 4.4.

The prototype implementation does not have any user interface. Instead, it has test scripts that simulate user activity using the API of the Process Insight Repository, as if a user was working with an application following use cases described in Section 1.2. These test scripts can be used as a how-to reference for development of a real user interface.

4.1. Components of the prototype implementation

The component structure of the prototype implementation follows reference component model described in Section 3.2.5 in Figure 24. The data of the Process Insight Repository is stored in a relational data model in IBM DB2 database version 9.7. IBM Data Studio version 3.1.1 was used to design the relational data model of the Process Insight Repository. The platform of the Process Insight Repository is implemented in Java Standard Edition 1.7 using Eclipse development environment. The component structure of the prototype implementation is shown in Figure 27.

An open source data mining environment RapidMiner version 5.2 (described in Section 3.2.5) was selected for the role of the data mining tool to perform data mining tasks in the Process Insight Repository. The API of RapidMiner is used to integrate it into the Process Insight Repository platform (see Section 4.3 for details). RapidMiner PMML extension is used for storing and reusing data mining models in PMML format (described in Section 3.2.2).

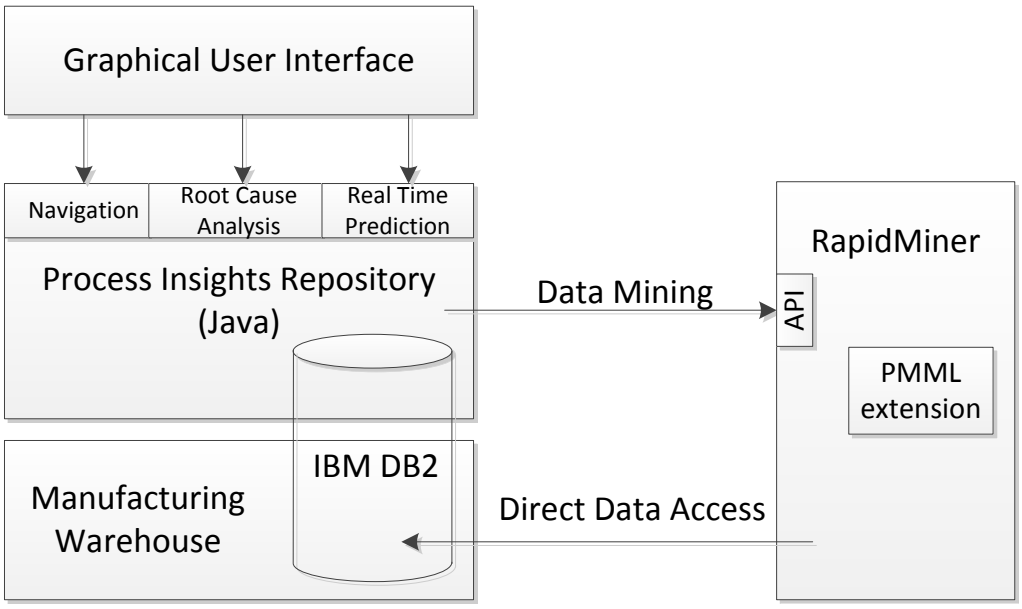

Figure 27: Component structure of the prototype implementation of the Process Insight Repository

4.2. Schema of the data model for the Prototype Implementation

The conceptual schema of the Process Insight Repository described in Chapter 2 contains the complete information about a manufacturing process. However, there is no sample data of a manufacturing process available for testing of the prototype implementation; therefore the schema of the manufacturing process model was significantly simplified for the prototype implementation. The simplified schema is presented in Figure 28. It has the main parts of the manufacturing process model as described in Section 2.1: Manufacturing Process and Manufacturing Process Instance, Production Step and Production Step Instance and a simplified version of Material Gateways with Input Materials left out. The number of Production Step dimensions was reduced to Machine, Employee and Fault dimensions (see Figure 28). All the parts of the schema are modeled as a relational database and stored in IBM DB2 RDBMS. IBM Data Studio was used to model the schema for the prototype implementation.

A new entity was added to the schema of the Process Insight Repository in order to store the data mining process. A Data Mining Process represents a prepared RapidMiner process that is used to train new Data Mining Models. Hence, each Data Mining Model has an associated Data Mining Process that was used to create it. Data Mining Process is described in details in Section 4.3.

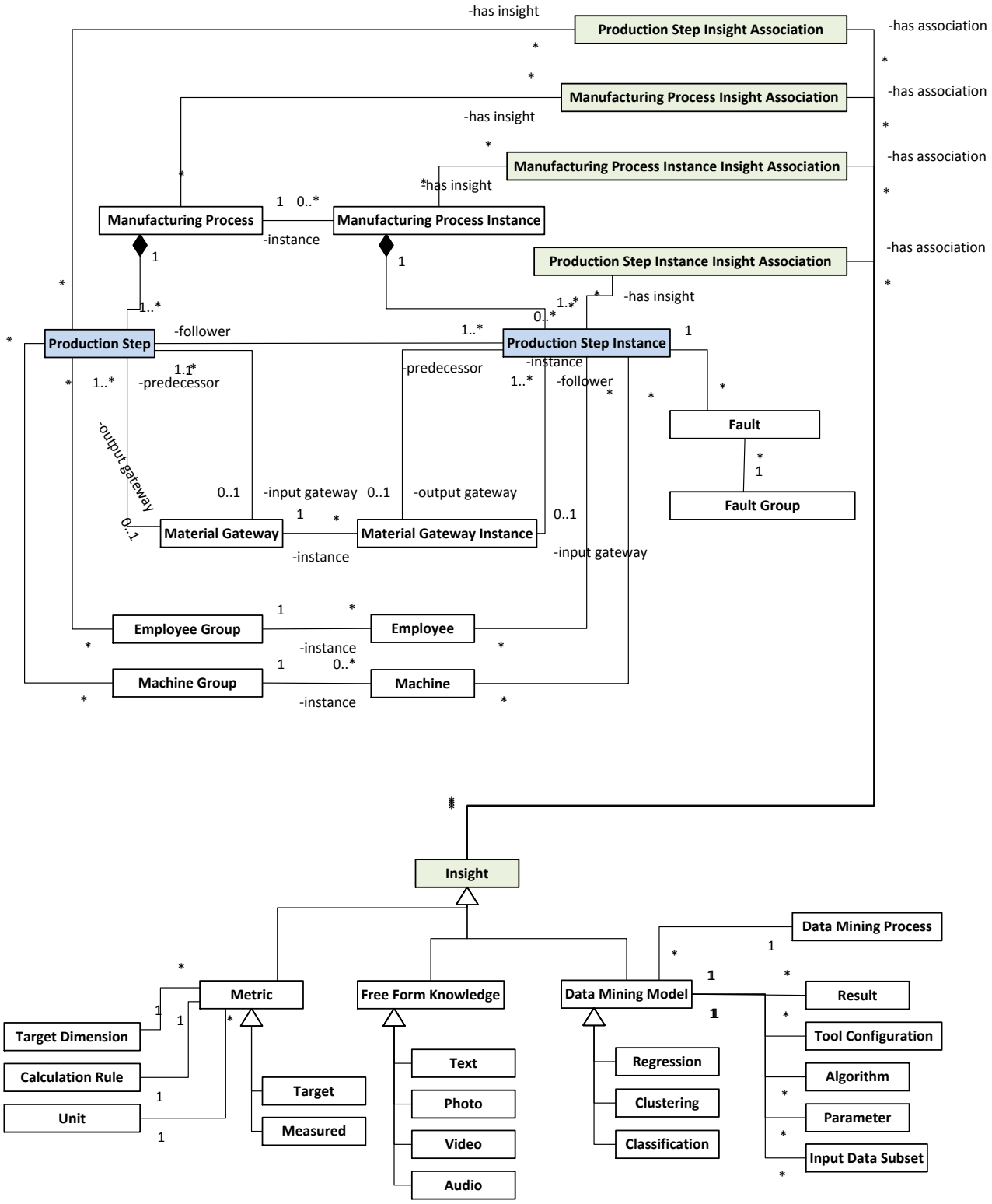

Figure 28: Simplified schema for the prototype implementation of the Process Insight Repository

4.2.1. Polymorphic Insight Associations

The schema of the Process Insight Repository contains many-to-many associations between entities of the manufacturing process model and different types of insights. The implementation of such associations is problematic, because different types of insights can be associated with different entities of the manufacturing process model. Such associations are called polymorphic associations, since they can associate objects of different classes.

Two approaches were used to resolve this problem in the prototype implementation of the Process Insight Repository. In the first place, different types of insights were generalized by the Insight entity using the base parent table approach described in [57]. The Insight table identifies insights of all kinds and holds data that is the same for all kinds of insights, e.g. ID of the insight, name, description, date of creation. These relationships are shown in Figure 29.

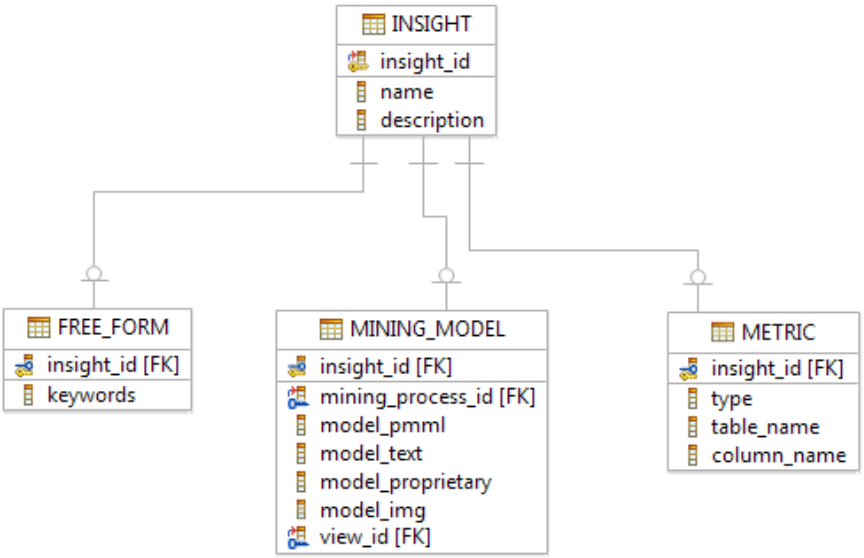

Figure 29: Generalized Insight entity

In the second place, entities of the manufacturing process model were associated with the generalized Insight entity using reverse relationship approach described in [57]. This approach has a benefit of having foreign key relationship constraints compared to other solutions of polymorphic associations in relational databases. Implementation of these relationships in the prototype of the Process Insight Repository is shown in Figure 30.

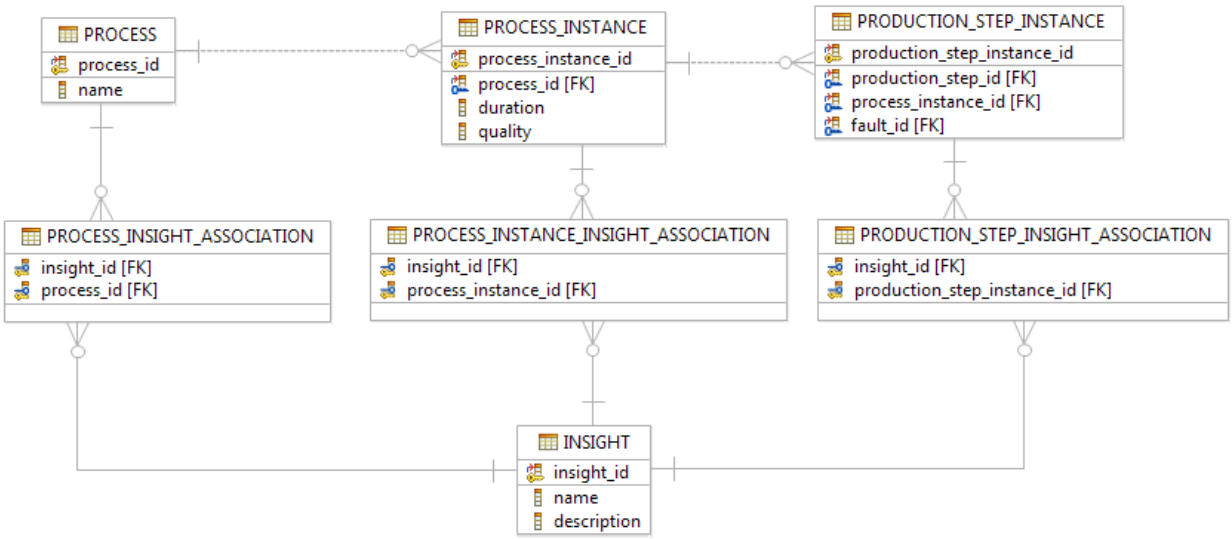

Figure 30: Associations with entities of the manufacturing process model

4.3. Integration of RapidMiner

The data mining tool that is used in the prototype implementation, RapidMiner, provides rich API, so that almost any task that can be performed in the graphical user interface of RapidMiner can be also performed programmatically using the provided API. RapidMiner uses the term "process" which represents a data mining process. RapidMiner has a set of predefined operators that can perform various tasks, e.g. read data from database, train a Bayes classifier or export created data mining model. Each operator can have multiple inputs and outputs, so that the output of one operator can be directed into input of another operator. A data mining process is a set of operators selected by the user and connected with each other [58]. An example of a RapidMiner process is given in Figure 31. A RapidMiner process can be exported and imported in an XML-based format.

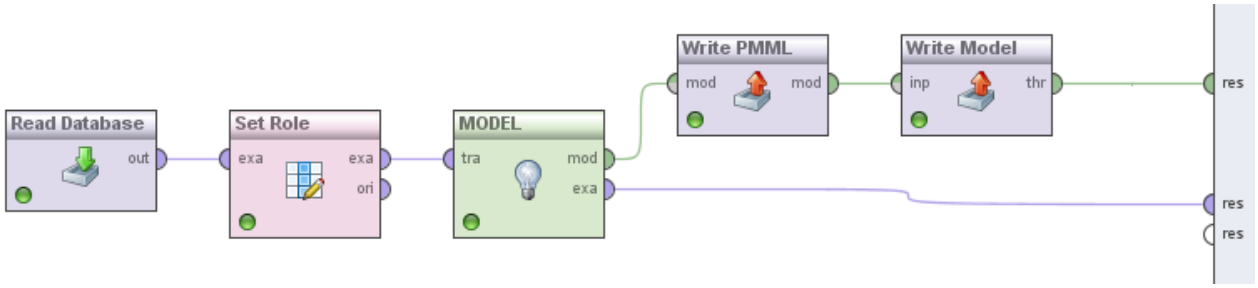

Figure 31: Example of a RapidMiner process for training a data mining model

There are two options to integrate RapidMiner into the Process Insight Repository. The first option is to create a new RapidMiner process programmatically. This means that the Process Insight Repository must use RapidMiner API to initialize a new RapidMiner process, add and connect operators, set their properties, then execute the created process and get the results. This approach is possible to implement, however, it showed to be error-prone and difficult for testing and debugging.

The second option of integrating RapidMiner showed to be more effective and allowed integration of custom RapidMiner processes into the Process Insight Repository (see Section 4.6.4 for details). This option involves manual creation of a RapidMiner process using the RapidMiner GUI and storing it into the Process Insight Repository. It allows performing the preparation of the data mining process by hand, thus allowing more options of RapidMiner operators to be tested. The RapidMiner process created manually is then exported in an XML-based format and stored in the Process Insight Repository. This eliminates the need for the Process Insight Repository to create a new RapidMiner process programmatically from scratch. Instead, RapidMiner is initialized with a prepared RapidMiner process and the only tasks that need to be performed programmatically is to change the settings of some operators, execute the process and read the results of process execution.

The operators that need to be setup are the "Read Database" operator that executes a SQL-statement in the DB2 database. Thus, a correct database connection to the Process Insight Repository database must be setup and the prepared SQL statement provided. In order to specify roles of attributes in the resulting dataset, the "Set Role" operator must be setup. In order to setup the "Set Role" operator, the Process Insight Repository retrieves the information about attributes from the column metadata (see Section 4.5.1 for details) and specifies how to treat different columns of the input dataset in the "Set Role" operator. Extended parameters of the data mining operators can also be programmatically adjusted at this stage if requested by the user of the Process Insight Repository.

After the RapidMiner process setup stage is completed, the process can be executed and the results of the data mining process retrieved by the Process Insight Repository. The result of the RapidMiner process execution can be of any form that is supported by RapidMiner, e.g. a data mining model in RapidMiner XML-base format or PMML format, or a dataset with predicted values. Hence, the Process Insight Repository must know how to treat the results that are received after executing a RapidMiner process. In order to implement this, two types of RapidMiner processes were defined to be used for the Process Insight Repository based on the need of use cases described in Section 1.2:

- Training of a data mining model on the selected data.
- Application of a data mining model to the selected data.

These two types of RapidMiner processes are described in details in the two following sections.

4.3.1. Training of a data mining model on the selected data

This type of RapidMiner process is used to create new data mining models based on the data selected by the user. An example of such RapidMiner process for training a new data mining model is given in Figure 31. The input of this RapidMiner process is a dataset prepared by the Process Insight Repository in a form of an SQL-statement. The output of this RapidMiner process is a data mining model in several formats. This data mining model can be directly represented to the user and stored in the Process Insight Repository for reuse. The data mining model is received from a RapidMiner process in several formats and all of them are stored in the Process Insight Repository. Two formats are stored for later automatic reuse – RapidMiner

XML-based format and PMML format; two formats are stored for the user in a representative form – a text description of the data mining model and graphical representation of the data mining model rendered by RapidMiner (an example of a decision tree rendered by RapidMiner is given in Figure 32).

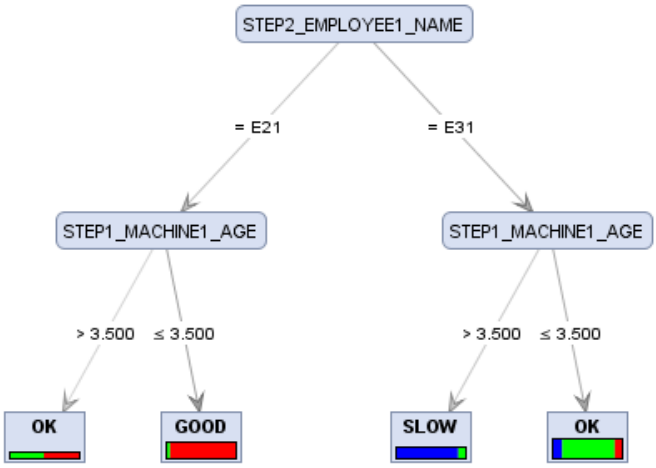

Figure 32: Example of a decision tree rendered by RapidMiner

This type of RapidMiner process is prepared by a data mining specialist and uploaded to the Process Insight Repository to be used by other users of the Process Insight Repository. Since this type of RapidMiner process will be used automatically and the Process Insight Repository must modify settings of its operators as described above, there are a few requirements for this type of RapidMiner process. A RapidMiner process for training of data mining models prepared by a data mining specialist must satisfy the following requirements:

- The operator that receives the input data subset must be named “Read Database”.
- The output of the “Read Database” operator must be passed through a “Set Role” operator named “Set Role”. The “Set Role” operator does not have to directly follow the “Read Database” operator, but it must exist on the path of the data from the “Read Database” operator to the data mining operator in order to assign correct roles for the attributes of the dataset.
- The operator that outputs the data mining model in RapidMiner XML format must be named “Write Model”.
- The operator that outputs the data mining model in PMML format must be named “Write PMML”.

It is the responsibility of the data mining specialist to prepare a RapidMiner process with the above mentioned operators named in the specified way. The conformity of the prepared RapidMiner process can be limitedly checked by the Process Insight Repository by executing the RapidMiner process on a sample set of data. As for the rest of the RapidMiner process, the data mining specialist has full freedom to use any operators provided by RapidMiner, i.e. any data filtering or data mining operators and any pre-processing or post-processing operators can be used in a prepared RapidMiner process.

4.3.2. Application of a data mining model to the selected data

The second type of a RapidMiner process that is defined for the Process Insight Repository applies the provided data mining model to the data selected by the user. This type of RapidMiner process is predefined in the Process Insight Repository platform, as it does not have any settings and only needs a prepared data mining model and input data. This RapidMiner process is shown in Figure 33.

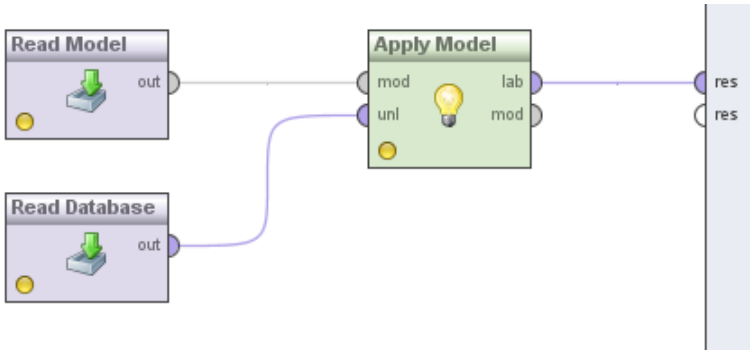

Figure 33: RapidMiner process for application of a data mining model to the new data

The input of this RapidMiner process is a data mining model in PMML format that was previously stored in the Process Insight Repository and a dataset selected by the user and prepared by the Process Insight Repository. The output of this RapidMiner process is a dataset with results of applying the data mining model to the input data. This result is retrieved by the Process Insight Repository from the RapidMiner process and represented directly to the user. The result can be also stored in the Process Insight Repository and associated with the corresponding entity of the manufacturing process model, e.g. the predicted classification label can be stored and associated with the Manufacturing Process Instance (described in Section 2.1.1) for which it was predicted.

4.4. Sample Manufacturing Process

A model of a manufacturing process can have different problematic kinds of data that have to be properly treated when preparing the data for a data mining process. Examples of such problematic data are conditional execution of production steps, repetitive execution of production steps or parallel execution of some part of a manufacturing process. For the purposes of testing the prototype implementation of the Process Insight Repository, a sample manufacturing process of producing steel-alloy coiled springs was taken as an example. The sample manufacturing process consists of the four following steps [59]:

1. Winding – a cold wire is wound around a shaft using a spring-winding machine.
2. Tempering – the spring is tempered by heat treating it. This step is done in order to relieve the stress in the material created in the winding step.
3. Shot peening – a tempered spring is strengthened by exposing the entire surface of the spring to a barrage of tiny steel balls that hammer it smooth and compress the steel that lies just below the surface.

4. Testing – a quality control is performed in order to check that the produced spring corresponds to the quality requirements.

The sample manufacturing process of producing coiled springs is taken in a simplified version. However, this simplified version is enough to cover several problematic test cases of manufacturing process data, such as repetitive production steps or uncompleted process instances.

4.4.1. Generation of sample manufacturing process data

As far as there is no sample manufacturing process data available for testing of the prototype implementation, the data of the manufacturing process has to be generated artificially. Hence, a data generator was implemented to populate the manufacturing process data in the Process Insight Repository. The sample manufacturing process data is generated in two steps.

First, a build-time model of the process is created, i.e. the Manufacturing Process described above consisting of the Production Steps connected with Material Gateways. Then resources that are used for this manufacturing process, such as Machines and Employees, are created in the build-time model and associated with the corresponding Production Steps.

After the build-time model of the sample spring winding process is created, the run-time model of the manufacturing process is populated with executions of the spring winding process, i.e. a number of Manufacturing Process Instances are created, each consisting of Production Step Instances. The run-time data of the sample manufacturing process is filled with random metrics data, e.g. random execution durations and random faults.

In order for the sample data to be usable for data mining tasks, it is generated in such way that metrics of Manufacturing Process Instances are dependent on resources that were used in a particular Manufacturing Process Instance. For example, participation of particular Employees result in shorter durations and usage of some Machines result in higher fault probability. These correlations are necessary for testing data mining use cases of the Process Insight Repository described in Section 1.2.

4.5. Data Preparation

Data preparation is an important step of preparing data selected by the user for the data mining tool. The data of a manufacturing process model is stored in a normalized form in the Process Insight Repository. However, the data mining tools expects the data to be in a one-row-per-subject format. Depending on the subject of analysis, the subject can be a Manufacturing Process Instance, Production Step Instance or Operating Resource usage. It is the task of the Process Insight Repository to transform the selected data to the format that can be used by a data mining tool. This kind of data transformation has many names: transposing, denormalizing, pivoting and others [60].

In order to do the data preparation in the Process Insight Repository, a helper pivot method was implemented, that takes a dataset selected by a user and performs the necessary transformations of data using the metadata about the manufacturing process model. The

metadata of the manufacturing process model and is described in the following Section 4.5.1. The details of the helper pivot method are given in Section 4.5.2.

4.5.1. Manufacturing process model metadata

Entity metadata

One of the requirements for the implementation of the Process Insight Repository is that the model of a manufacturing process is flexible and must allow adding new dimensions. In order to handle these changes, the Process Insight Repository stores metadata describing the schema of the manufacturing process model. This metadata contains a list of entities of the manufacturing process model and relations between them. This metadata is used for the preparation of data for data mining tasks (see Section 4.5 for details).

Column metadata

Entities of the process model hold various collected data about manufacturing processes executed on a factory. Type of data that is stored in the column must be properly handled when this data is used in a data mining process (see Section 4.3 for detailed description of a data mining process). In order to do this, the Process Insight Repository stores metadata that describes the columns of the manufacturing process model and specifies how to use them. The Process Insight Repository has some default assumptions about data columns of the manufacturing process model. Hence, it is not necessary to describe every column of the data model with metadata. However, in the cases mentioned further in this section some columns must be described with metadata.

The first problem with treating data columns is to properly define data type of the column and correctly specify this information for the data mining tool. By default, the data type that is specified in the database is taken, i.e. types VARCHAR and CHAR are treated as nominal values and types INTEGER and REAL are treated as numerical values. However, there are numerical attributes that must be treated as nominal values, e.g. attribute “code of failure” can be stored as an integer number with value “0” meaning ”no errors”, value “1” meaning ”error type 1” and value “2” meaning “error type 2”. Such columns are described as nominal attributes in the column “column_type” of the table “Column Data Type” (see Figure 34). A data column is identified by the table name and the column name. In order to improve semantic representation of the data to the user, each value of such column can be assigned a label using the table “Column Value Label” in (see Figure 34).

The second problem with treating data columns is that some numeric columns must be discretized, e.g. an attribute “duration” of a Manufacturing Process Instance can be discretized to three categories – “slow”, “normal” and “fast”. As far as different Manufacturing Processes must be discretized individually, discretization is associated with a Manufacturing Process. For these columns, a set of ranges must be defined and each range is given a label. E.g., X < 4 means “FAST”, 4 <= X < 7 means “OK” and 7 <= X means “SLOW”.

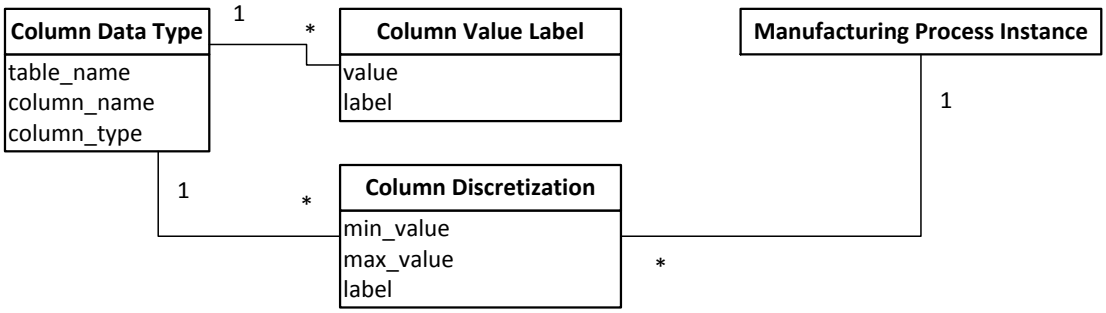

Figure 34: Column metadata diagram

The third problem is that some data columns must be omitted when data is prepared for data mining. ID columns must be omitted, because data mining must be based on object properties, but not object IDs. Measured Metrics columns must be omitted from the data prepared for data mining except for the Metrics that are the target of data mining task. For example, in the Root Cause Analysis use case (see Section 1.2) when we want to analyze the factors influencing the duration of the Manufacturing Process Instance execution, we must exclude durations of Production Step Instances from the data, because they are directly correlated with the duration of the Manufacturing Process Instance, but are out of interest in terms of the analysis. Hence, the table "Column Data Type" contains records for ID columns and data columns with Measured Metrics values. These records have the value of column_type set to "id" and "metrics" respectively. Data columns described by these records are excluded in the data preparation step so that this data will not be considered by the data mining tool when creating a new data mining model.

4.5.2. Pivoting data

In order to prepare the data of the manufacturing process for the data mining tool, it must be pivoted, i.e. represented in a one-row-per-subject form. This is done by the Process Insight Repository using a helper pivot method. An example of unprepared manufacturing process data is given in Table 7. In this case the subject of analysis is a Manufacturing Process Instance (P1 and P2 in the given example). Hence, a data mining tool expects to receive one row of data per Manufacturing Process Instance. However, the data is spread across multiple rows. The goal of data pivoting is to transform the given data to a one-row-per-subject form, as shown in Table 6.

As far as the model of a manufacturing process allows changes, SQL statements are not hardcoded, but generated using the manufacturing process metadata described in Section 4.5.1. In order to select all the available manufacturing process data, a FROM-clause is generated using the metadata and the filter provided by a user, e.g. to prepare data for only one particular Manufacturing Process. This FROM-statement joins all the tables of the manufacturing process model according to the metadata and is used in all the following steps of the pivoting process.

The first step of data pivoting is getting all the columns that are available in the manufacturing process model. These are the columns that are shown in the example of unprepared data in Table 7. In order to do this, a "SELECT * "-query is generated using the prepared FROM-clause.

This query fetches only the first row. Hence, no data is retrieved, but only the metadata of the query is used to determine all the columns that are available in the manufacturing process model. Each column of this query is treated according to the available column metadata described in Section 4.5.1, i.e. primary and foreign key columns are skipped, columns that need discretization are discretized, metrics are omitted except for the target metric that needs to be analyzed in the data mining process. Discretization of columns is done using the CASE WHEN statement, hence, the discretization parameters influence only when the data is prepared for the first time, but not later if they change. The set of columns is saved

Table 7: Example of unprepared data of a manufacturing process

Process Instance ID	Step ID	Machine ID	Machine Age	Employee ID	Employee Group
P1	S1	M1	3		
P1	S2	M58	2		
P1	S3	M64	2		
P1	S4	M72	2	E31	G3
P2	S1	M2	6		
P2	S2	M58	2		
P2	S3	M64	2		
P2	S4	M72	2	E21	G4

Table 6: Example of pivoted data of a manufacturing process

	Step S1		Step S2		Step S3		Step S4			
Process Instance ID	Step S1 Machine ID	Step S1 Machine Age	Step S2 Machine ID	Step S2 Machine Age	Step S3 Machine ID	Step S3 Machine Age	Step S3 Machine ID	Step S3 Machine Age	Step S4 Employee ID	Step S4 Employee Group
P1	M1	3	M58	2	M64	2	M72	2	E31	G3
P2	M2	6	M58	2	M64	2	M72	2	E21	G4

The second step of data pivoting is to determine the denormalization groups of data that must be transformed from rows to columns. In the given example of unprepared data in Table 7 these denormalization groups are shown with different colors, i.e. each Production Step forms a denormalization group.

The third step of data pivoting is to create a final SELECT statement that selects the set of columns defined in the first step for each denormalization group defined in the second step. Hence, the number of columns in the final SELECT statement is equal to the number of data columns multiplied by the number of denormalization groups. An example of data returned by the generated SQL query is shown in the Table 6.

The generated SELECT statement is used to create a view in the database, so that the results of the pivoting process could be reused. The name of the created view is stored in the metadata of the Process Insight Repository, so that the view can be accessed any time later. The view provides data in a one-row-per-subject form. Therefore, it can be used as a data source by a data mining tool directly without any other preparations.

Multiple occurrence of Production Step in a Manufacturing Process Instance

The model of a manufacturing process might allow a production step to be executed more than one time, e.g. when a part must be reworked after quality check. Let us consider an example of the sample Manufacturing Process described in Section 4.4 with three Manufacturing Process Instances, one of which has Production Step 3 (Shot peening) repeated three times. As far as each Production Step becomes a set of columns in pivoted prepared data, there are several ways to treat multiple Production Step Instances of a Production Steps.

The trivial solution is to include each occurrence of a Production Step as a separate set of columns, as shown in Table 8. This approach has several drawbacks. One of the drawbacks is that most of the rows of the prepared data will have null values in repetitive columns (rows of Process Instances P1 and P3) and only rows that correspond to the Manufacturing Process Instances that have repetitive Production Steps will have data in these columns (row of Process Instance P2). This reduces the understandability of the data by the user, i.e. the user will have to see more columns of data and understand the meaning of data in such columns. Another drawback is that it is hard to know the number of Production Step occurrences in advance, i.e. there can be Manufacturing Process Instances with three or nine occurrences.

Table 8: Example 1 of multiple Production Step occurrence

Process Instance ID	…	Step3 Instance1 Machine1 ID	Step3 Instance1 Machine1 Age	Step3 Instance2 Machine1 ID	Step3 Instance2 Machine1 Age	Step3 Instance3 Machine1 ID	Step3 Instance3 Machine1 Age	…
P1		M1	3					
P2		M2	6	M2	6	M3	3	
P3		M2	6					

Another solution to the problem of repetitive Production Steps is to include only one Production Step Instance of each Production Step. This will lead to the loss of source information for the data mining tool to some extent, but improve the understandability of the data by the user. As far as only one Production Step Instance must be included for each Production Step, the data preparation algorithm can take either the first or the last instance of a Production Step. To indicate the fact that a Production Step was executed more than one time in the data for the data mining tool, the number of Production Step occurrences can be included for each Manufacturing Process Instance. An example of the data prepared with this approach is shown in Table 9.

Table 9: Example 2 of multiple Production Step occurrence

Process Instance ID	…	Step3 Instance1 Machine1 ID	Step3 Instance1 Machine1 Age	Step3 Instance1 count	…
P1		M1	3	1	
P2		M2	6	3	
P3		M2	6	1	

4.6. Implementation of sample use cases

4.6.1. Navigation of the Process Insight Repository

In the navigation use case a user can browse the Process Insight Repository, i.e. search and observe objects of the manufacturing process model and retrieve insights that are associated with them. This is implemented in the prototype by representing entities of the Process Insight Repository with Java classes, i.e. each entity that is presented in Figure 28 has an associated Java class. The Process Insight Repository provides a navigation API that allows the user to search and retrieve objects of the Process Insight Repository. For example, the user can find and retrieve a particular Manufacturing Process and its build-time model consisting of Production Steps and associated resources; retrieve the Manufacturing Process Instances of this Manufacturing Process; find all available Insights that are associated with the selected Manufacturing Process.

4.6.2. Root Cause Analysis

The goal of the Root Cause Analysis technique is to find out how the parameters and properties of the selected Manufacturing Process influence the value of the selected Measured Metric, e.g. which parameters of a manufacturing process result in higher lead times. This problem can be solved by using classification data mining techniques that can be interpreted by the user [3]. Examples of such interpretable data mining techniques are decision tree induction and decision rules generation. In the prototype implementation of the Process Insight Repository the Root Cause Analysis use case is implemented with generation of a decision tree that classifies the Measured Metric specified by the user.

In this use case a user selects a Manufacturing Process and one of the available Measured Metrics using the navigation API of the Process Insight Repository. The selected Measured Metric must have the discretization setup (see Section 4.5.1). If the discretization was not setup before, the user can setup discretization parameters using the API described in Section 4.7.2. These are the input parameters for the Root Cause Analysis.

After the input parameters for the Root Cause Analysis were specified by the user, the Process Insight Repository prepares the data of the selected Manufacturing Process as was described in Section 4.5 and executes a predefined data mining process for the prepared data as was described in Section 4.3.1. The predefined data mining process uses the Decision Tree operator of RapidMiner to derive a new decision tree data mining model from the given input data. An

experienced user can also specify some parameters of the data mining algorithm, such as pruning of the created decision tree or data filter parameters.

The created data mining model is then stored as a new Insight in the Process Insight Repository in PMML format and is associated with the Manufacturing Process for which it was created. The user receives the created data mining model in one of the available forms, i.e. text form, rendered image or in PMML format.

4.6.3. Real Time Prediction

One of the use cases for prototype implementation of the Process Insight Repository is prediction of various metrics of a currently running manufacturing process instance [3]. In this use case there is a manufacturing process instance that has not completed yet, i.e. not all Production Steps have been executed. The Process Insight Repository already contains the data of the production steps that were completed in this process instance. A user wants to get the predicted value of a selected metric for the uncompleted manufacturing process instance. The PIR provides the required value by applying the most suitable data mining model from the set of trained data mining models for this metrics.

This use case introduces several problems that are solved in the prototype implementation of the PIR. One of the problems is that the data of completed manufacturing process instances contains all the production steps that were executed, whereas the currently running process instance contains only those production step instances that were already executed. Thus, if we train a data mining model on all of the available data, it will not be usable for the currently running process, because it will have missing values for uncompleted production steps.

In the perfect case the PIR should have a data mining model that is trained using exactly the same production steps that are completed by the current process. A straight solution to this problem is to train the model on the same set of production steps that are completed by the current process instance. However, if the amount of accumulated process data is big, training a model might take too long time. Long training time makes this solution inappropriate for using in real time prediction, as the user expects to receive the predicted value of the metric in some reasonable time. To achieve short response time for the user, the PIR must already have a set of trained data mining models. These data mining models must be trained in advance, for example on a nightly or weekly basis, so that when the user requests prediction, the PIR would only have to apply a ready-made data mining model to the data of the currently running process instance.

The perfect data mining model would include all the completed steps of the uncompleted process instance. Let us consider a simple sequential process, it can be found in the following states: $C_1 = \{S_1\}$; $C_2 = \{S_1, S_2\}$; $C_3 = \{S_1, S_2, S_3\}$; $C_i = \{S_1 .. S_i\}$ and $C_N = \{S_1 .. S_N\}$.

Where S_i is the Production Step i of the Manufacturing Process and C_i is a set of completed Production Steps and represents the state of the process (see Figure 35). For a sequential process of N steps, there can be N states of the process instance. Thus, N data mining models are needed to provide the best prediction of metrics, with each model trained on a set of steps

C_i. However, a process might have many steps and data mining models trained on sets C_i and C_{i+x}, where x is a small positive number, can be almost the same and the model M_{i+x} might have small improvement of prediction compared to M_i. As training of data mining models consumes computational resources, optimization can be made by skipping some of the models. There are several possibilities to select states of the process for creation of data mining models:

- Select each Kth state, for example C_k, C_{2k}, C_{3k}, …, C_{j*k}, …, C_{N-k}, C_N, with j mod k=0. This approach is straight forward and doesn't require complicated setup by the user. The user must only specify the number of data mining models that need to be trained. This method has an obvious shortcoming, that the selection of states can be not optimal.
- Manual specification of steps by the user.
- Automatic selection of states can be done based on the weights of attributes of each production step. The states, where the attributes of the last production step have low weights in the data mining model, can be omitted.

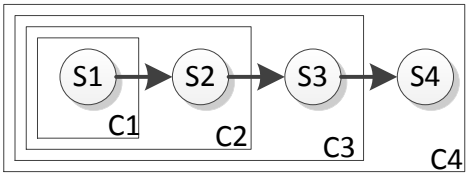

Figure 35: Example of uncompleted states of a Manufacturing Process

To implement automatic selection of process states for data mining models, the training of data mining models can be split into two phases. The first phase, selection of states, is performed only once to determine which states contribute the most to the prediction of selected metrics. At this step, data mining models are trained for each process state. In the second phase only the data mining models for the states selected in the first phase are updated. The second phase can be performed on a regular basis in order to update the data mining models that are used for real time prediction.

A Process can be seen as a graph with production steps as nodes and gateways between steps (see Section 2.1.3) as ribs. The execution of a process instance builds the graph of the process instance by adding a node with the termination of each production step instance.

For processes which have branches and parallel steps, the selection of steps becomes more complex. If the process graph contains production steps that are executed in parallel, the number of possible states of the process instance increases greatly compared with a sequential process.

4.6.4. Custom Data Mining Process

Two previous sections describe the implementation of the predefined use cases. These use cases use predefined data mining processes, are hardcoded and provide an API for convenience of the application developer. However, the designed prototype of the Process Insight Repository allows handling of newly added custom data mining processes without the need to

modify the source code of the Process Insight Repository platform. This functionality implements the custom data mining process use case by storing the complete custom data mining process from the data mining tool in the Process Insight Repository and making it available to users.

The value of this functionality is that compared to the predefined data mining use cases (Root Cause Analysis and Real-Time Prediction) it allows easy introduction of new data mining use cases without changing the source code or API of the Process Insight Repository. This allows the data mining specialist to easily define practically any data mining process that can be performed with RapidMiner and make it available for advanced users and non-expert users. When such custom data mining process is stored in the Process Insight Repository, its usage does not require any specific data mining knowledge from the users of the Process Insight Repository (advanced users and non-expert users).

In order to create a new data mining process in the Process Insight Repository, the following steps must be performed by the data mining specialist:

1. Prepare data for data mining using an interface to the Process Insight Repository. A view with denormalized data will be created in the database by the Process Insight Repository. All the information to access this view will be provided to the data mining specialist, including a connection string that can be directly used in RapidMiner to access the view in the database. This data is used only for the creation of the data mining process; after the data mining process will be created and uploaded, it can be applied to any other data by the Process Insight Repository.
2. Create a new data mining process in RapidMiner and export it using RapidMiner XML format for data mining process exchange.
3. Upload the exported data mining process to the Process Insight Repository using the corresponding API (see Section 4.7.5).

After performing these steps, the Process Insight Repository contains a newly added data mining process, which is available with the custom data mining process API (see Section 4.7.5). The implementation of the Process Insight Repository allows using such data mining processes not only for the data on which they were created, but for any data that can be prepared by the Process Insight Repository.

In order to use such custom data mining process, the application for the advanced user must use the custom data mining process API of the Process Insight Repository. In order to use a custom data mining process, advanced user must perform the following steps:

1. Select a custom data mining process using the extended navigation API of the PIR.
2. Select the data and the target attribute for application of a custom data mining process using the navigation API of the PIR.
3. Create a new Data Mining Model by providing the selected data mining process and the selected data to the API call of the PIR (see Section 4.7.5). A new Data Mining Model will be created and associated with the source data and the custom data mining process.

Advanced user can inspect the Data Mining Model if a descriptive data mining technique is used, or store the Data Mining Model in the Process Insight Repository to make it ready for later use.

The PIR encapsulates performing of a data mining task in the step 3. In order to do it, the PIR calls RapidMiner and initializes a data mining process using the stored data mining process selected by the advanced user. The operators of the process are altered for the data selected by the user, i.e. the source of data is modified to match the data selected by the user and the selected target attribute is assigned with the target role.

If the advanced user stored the newly created Data Mining Model in the PIR, it can be applied to the new data. In order to apply a custom Data Mining Model to the new data, the following steps must be performed by the advanced user:

1. Select the Data Mining Model using extended navigation API of the PIR.
2. Select new data from the data that matches the predicate filter of the Data Mining Model (see Input Data Subset in Section 2.2.1). The PIR provides an API call to search the data that matches a selected Data Mining Model.
3. Apply the selected Data Mining Model to the selected data using an API call. The PIR will return the result of applying the Data Mining Model to the user.

The same way as in the creation of a new Data Mining Model, the PIR encapsulates application of a Data Mining Model to the new data. This is done by calling RapidMiner and initializing a new data mining process using the data selected by the user and the selected Data Mining Model.

4.7. API of the Process Insight Repository

The Process Insight Repository is a part of the Advanced Manufacturing Analytics platform and therefore must provide an Application Programming Interface (API) for the higher levels of the platform. For the simplicity of application development, the Process Insight Repository provides the functionality for the described use cases as simple API calls. The API of each use case is implemented as a class that exports only the methods that are necessary for this use case. Hence, the API of the Process Insight Repository can be classified by use cases. The further subsections summarize the API provided by the Process Insight Repository.

4.7.1. Navigation of the Process Insight Repository

The navigation API provides calls to navigate the manufacturing process model and associated insights in the Process Insight Repository. This API is inherited by other use case APIs, thus this is the basic functionality that is available in every use case. With the help of Navigation API a user can navigate the Process Insight Repository. Following are examples of calls provided by this API:

- Search for entities of the model, e.g. available Manufacturing Processes and Manufacturing Process Instances, Resources and Employees.
- Retrieve Insights associated with a selected object of the manufacturing process model, e.g. retrieve insights into the Manufacturing Process “Spring Winding”.

- Search Insights using a search string given by the user. This functionality can be used for retrieving free form knowledge available in the Process Insight Repository.

4.7.2. Extended API for the advanced user

This API is available only to the advanced user and the data mining specialist. It is used to setup some parameters of the Process Insight Repository.

- Create new Insights, e.g. upload new Free Form Knowledge documents.
- Setup discretization of the selected Metric for the selected Manufacturing Process. This call is used to setup the discretization of metrics by specifying value intervals and corresponding labels (see Section 4.5.1).
- Setup nominal labels for an attribute. This call is used to setup numeric attributes that must be treated as nominal, e.g. “0” means “no error” and “1” means “fault”.
- Create association between an insight and an object of the manufacturing process model, i.e. associate a Free Form Knowledge with a particular Machine Group.

4.7.3. Root Cause Analysis API

This API provides a single call to create a decision tree:

- Create a decision tree for the specified Manufacturing Process and the Metric selected by the user. This call performs the steps described in Section 4.6.2 and returns a derived data mining model which can be presented to the user in one of the available forms (text, image or PMML). The advanced user can specify additional parameters for data mining process, e.g. disable pruning of the generated decision tree.

4.7.4. Real Time Prediction

This API provides calls for the Real Time Prediction use cases:

- Train prediction data mining models for the specified Manufacturing Process and the Metric. Within this call the Process Insight Repository trains data mining models as described in Section 4.6.3 based on the data of completed Manufacturing Process Instances. This call can be used to train data mining models for prediction in advance, e.g. nightly or hourly.
- Predict a value of the selected Metric for the selected uncompleted Manufacturing Process. Within this call the Process Insight Repository find the most suitable data mining model and applies it to the data of the uncompleted Manufacturing Process. Training of prediction data mining models is invoked in some cases, i.e. when no data mining models are available, if available data mining models are expired or if requested by the user.

4.7.5. Custom Data Mining Process

This API enables easy deployment of new data mining processes by the data mining specialist and their usage by advanced users (see Section 4.6.4). As far as these two types of users have different expertize in data mining, the API is split into two parts – one for each kind of user.

API for the data mining specialist

With the help of this API a data mining specialist can prepare data for performing data mining with external data mining tools and deploy new data mining processes into the Process Insight Repository. Following are the calls provided by this API:

- Prepare the selected data for data mining. This call performs data preparation as described in Section 4.5.2 and creates a view on the prepared data in the database. The created view contains the prepared data and can be used directly from an external data mining tool.
- Upload new custom data mining process. This call uploads an XML file with the new data mining process created in RapidMiner as described in Section 4.6.4.

API for the advanced user

With the help of this API an advanced user can create and apply new data mining models using custom data mining processes created by the data mining specialist.

- Get list of custom data mining processes.
- Create a new data mining model for the selected Metric and selected Manufacturing Process using the selected data mining process. This call creates a new data mining model and stores it in the Process Insight Repository.
- Apply a custom data mining model to the new data. This call can be used to apply a data mining model created with a custom data mining process to the new data.

4.7.6. Altering the manufacturing process model

One of the requirements to the Process Insight Repository is to allow changes in the model of a manufacturing process [5]. Section 4.5.1 described the metadata that is needed to handle the structure of the manufacturing process model dynamically. The Process Insight Repository must as well provide an API that allows changing this manufacturing process metadata. With the help of this API the manufacturing process model can be altered in the Process Insight Repository. This API must provide calls to extend the Process Insight Repository with new entities of the manufacturing process model and describe usage of attributes of the manufacturing process model. Examples of calls provided by this API are:

- Add entities to the manufacturing process model and specify their relations to existing entities.
- Add attributes to the existing dimensions and specify their usage.

5 Conclusion

In terms of this work a general concept of the Process Insight Repository was developed and a prototype implemented. The Process Insight Repository is a part of the Advanced Manufacturing Analytics (AdMA) platform, therefore its goal is to ensure continuous improvement of a manufacturing process by providing an integrated storage for the insights into the manufacturing process. The Process Insight Repository enables storing various kinds of insights associated with the manufacturing process data, so that the user of the platform can access the accumulated knowledge about manufacturing processes. Systematic storing of insights is valuable for process improvement, because the knowledge derived from the manufacturing process data is not used only once and lost, but is stored and can be accessed in the repository.

This work showed that data mining techniques can be integrated into the Process Insight Repository in a way that does not require specific knowledge from the user. This allows application of data mining techniques to the manufacturing process data not only by data mining specialists, but also by simple users. Hence, users of the platform can discover valuable knowledge in the manufacturing process. The prototype implementation also showed that the Process Insight Repository allows simple deployment of new custom data mining processes without the need to change the source code or the API of the platform. Hence, the Process Insight Repository can be used as a platform for easy deployment of data mining processes. This also allows easy extension of the Process Insight Repository with new data mining use cases.

The design of the conceptual schema of the Process Insight Repository showed that there are different options to model manufacturing processes. The model of the manufacturing process mainly depends on the data that is available. Therefore the prototype was implemented in a way that allows changes in the model of the manufacturing process. This allows storing all the available manufacturing process data regardless of its structure.

The future work that can be done with regard to the Process Insight Repository includes:

- Develop ETL (Extract, Transform and Load) procedures to populate the Process Insight Repository with the real data that is available for the manufacturing process.
- Extend the Process Insight Repository with new data mining use cases.
- Integration of the Process Insight Repository with other parts of the Advanced Manufacturing Analytics platform.

References

[1] C. Gröger, J. Schlaudraff, F. Niedermann and B. Mitschang, "Warehousing Manufacturing Data - A Holistic Process Warehouse for Advanced Manufacturing Analytics.," in *Data Warehousing and Knowledge Discovery (DaWaK)*, Vienna, Austria, 2012.

[2] F. Niedermann and H. Schwarz, "Deep Business Optimization: Making Business Process Optimization Theory Work in Practice," in *Enterprise, Business-Process and Information Systems Modeling - 12th International Conference, BPMDS 2011, and 16th International Conference, EMMSAD 2011, held at CAiSE 2011, London, UK, June 20-21, 2011. Proceedings*, 2011.

[3] C. Gröger, F. Niedermann and B. Mitschang, "Data Mining-driven Manufacturing Process Optimization," in *Proceedings of the World Congress on Engineering 2012 Vol III*, London, U.K., 2012.

[4] C. Gröger, "Konzeption und prototypische Umsetzung einer Referenzarchitektur für Manufacturing Analytics," 2011.

[5] F. Niedermann, H. Schwarz and B. Mitschang, "Managing Insights: A Repository for Process Analytics, Optimization and Decision Support," in *Proceedings of the International Conference on Knowledge Management and Information Sharing (KMIS) 2011*, Paris, 2011.

[6] K. Erlach, Wertstromdesign. Der Weg zur Schlanken Fabrik, Berlin Heidelberg: Springer-Verlag, 2007.

[7] C. McLean and S. Leong, *A Process Model For Production System Engineering.*

[8] R. V. Rao, Advanced Modeling and Optimization of Manufacturing Processes, London: Springer-Verlag, 2011.

[9] *DIN EN 62264-1 (2008): Integration von Unternehmensführungs- und Leitsystemen. Teil 1: Modelle und Terminologie,* Deutsches Institut für Normung (DIN), 2008.

[10] S. Zor, F. Leymann and D. Schumm, "A Proposal of BPMN Extensions for the Manufacturing Domain," *Proceedings of the 44th CIRP International Conference on Manufacturing Systems,* 2011.

[11] B. Waddel, *Manufacturing's Five Golden Metrics,* 2006.

[12] ibk Industrieservice GmbH, "Datenbankbasierte Planung mit Process Designer," [Online]. Available: http://www.ibk-hannover.de/Leistungen/em-planner.html. [Accessed 05 11 2012].

[13] ProcessMaker, "ProcessMaker: Object Model for Process definition," [Online]. Available: http://wiki.processmaker.com/index.php/Object_Model_for_Process_definition. [Accessed 06 11 2012].

[14] ProcessMaker, "ProcessMaker Architecture Diagrams," [Online]. Available: http://wiki.processmaker.com/index.php/ProcessMaker_Architecture_Diagrams. [Accessed 06 11 2012].

[15] ProcessMaker, "ProcessMaker: Importing and Exporting Process," [Online]. Available: http://wiki.processmaker.com/index.php/2.0/Importing_and_Exporting_Process. [Accessed 06 12 2012].

[16] W. Terkaj, G. Pedrielli and M. Sacco, "Virtual Factory Data Model," [Online]. Available: http://kr-med.org/icbofois2012/proceedings/ICBOFOIS2012Workshops/FOIS2012OSEMA/FOIS-2012-OSEMA-Terkaj-1.pdf. [Accessed 05 11 2012].

[17] W. Terkaj and M. Urgo, "Virtual Factory Data Model to support Performance Evaluation of Production Systems," 2012. [Online]. Available: http://kr-med.org/icbofois2012/proceedings/ICBOFOIS2012Workshops/FOIS2012OSEMA/FOIS-2012-OSEMA-Terkaj-2.pdf. [Accessed 05 11 2012].

[18] W3C, "OWL Web Ontology Language Guide," 10 02 2004. [Online]. Available: http://www.w3.org/TR/owl-guide/. [Accessed 05 11 2012].

[19] "Poll: Data Mining Methodology (Aug 2007)," 08 2007. [Online]. Available: http://www.kdnuggets.com/polls/2007/data_mining_methodology.htm. [Accessed 16 09 2012].

[20] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," *AI Magazine,* vol. 17, no. 3, pp. 37-54, 1996.

[21] SAS, "SAS Enterprise Miner: SEMMA," SAS, [Online]. Available: http://www.sas.com/offices/europe/uk/technologies/analytics/datamining/miner/semma.html. [Accessed 02 12 2012].

[22] S. S. Rohanizadeh and M. B. Moghadam, "A Proposed Data Mining Methodology and its Application to Industrial Procedures".

[23] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer and R. Wirth, CRISP-DM 1.0. Step-by-step data mining guide, SPSS, 2000.

[24] A. Azevedo and M. F. Santos, "KDD, SEMMA and CRISP-DM: a parallel overview," *Proceedings of the IADIS European Conference on Data Mining 2008,* pp. 182-185, 2008.

[25] "Process Model," [Online]. Available: http://crispdm.wordpress.com/process-model/. [Accessed 16 09 2012].

[26] Data Mining Group, "PMML 4.1 - General Structure of a PMML Document," [Online]. Available: http://www.dmg.org/v4-1/GeneralStructure.html. [Accessed 15 11 2012].

[27] Object Management Group, Inc, "Common Warehouse Metamodel (CWM)," March 2003. [Online]. Available: http://www.omg.org/spec/CWM/. [Accessed 15 11 2012].

[28] A. Guazzelli, "What is PMML? Explore the power of predictive analytics and open standards (IBM)," IBM Corporation, 2010.

[29] "Data Mining Group - PMML Powered," [Online]. Available: http://www.dmg.org/products.html. [Accessed 27 08 2012].

[30] I. Ntoutsi and Y. Theodoridis, "Current Issues in Modeling Data Mining Processes and Results," *PANDA Workshop on Pattern-Base Management Systems, April, 10th 2003,* pp. 11-20, 2003.

[31] J. Darmont and O. Boussaïd, Processing And Managing Complex Data for Decision Support, Idea Group Inc, 2006.

[32] M. ANDRONIE and D. CRISAN, "Commercially Available Data Mining Tools used in the Economic Environment," *Database Systems Journal vol. I, no. 2/2010,* vol. 1, no. 2, pp. 45-54, 2010.

[33] KDnuggets, "Poll Results: analytics/data mining tools used for a real project," May 2011. [Online]. Available: http://www.kdnuggets.com/2011/05/tools-used-analytics-data-mining.html. [Accessed 16 09 2012].

[34] J. Melton and A. Eisenberg, "SQL Multimedia and Application Packages (SQL/MM)," *SIGMOD Rec.,* vol. 30, no. 4, pp. 97-102, 2001.

[35] A. Kadav, J. Kawale and P. Mitra, "Data Mining Standards," [Online]. Available: http://www.datamininggrid.org/wdat/works/att/standard01.content.08439.pdf. [Accessed 03 12 2012].

[36] Oracle, "Oracle Data Mining Techniques and Algorithms," [Online]. Available: http://www.oracle.com/technetwork/database/options/advanced-analytics/odm/odm-techniques-algorithms-097163.html. [Accessed 23 August 2012].

[37] Oracle, "Oracle Database 11g PL/SQL Packages and Types Reference: DBMS_DATA_MINING," 2011. [Online]. Available: http://docs.oracle.com/cd/E11882_01/appdev.112/e25788/d_datmin.htm. [Accessed 03 12 2012].

[38] IBM Corporation, "DB2 Business Intelligence / DWE Data Mining / Overview of the Data Mining features," [Online]. Available: http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=%2Fcom.ibm.im.overview.doc%2Fc_introducing_the_intelligent_miner_products.html. [Accessed 03 12 2012].

[39] "SQL Server Data Mining Programmability," [Online]. Available: http://msdn.microsoft.com/library/ms345148.aspx.

[40] "Migration Considerations (Analysis Services)," [Online]. Available: http://msdn.microsoft.com/en-us/library/ms143235%28v=sql.105%29.aspx.

[41] "Supported PMML Models," [Online]. Available: http://social.msdn.microsoft.com/Forums/en-US/sqldatamining/thread/11a8d222-ac2b-4843-99ac-b4a7ff65aadf/.

[42] IBM, "IBM SPSS Statistics Programmability Extension," [Online]. Available: http://www-142.ibm.com/software/products/us/en/spss-stats-programmability/. [Accessed 16 11 2012].

[43] IBM, "IBM SPSS Modeler Premium," [Online]. Available: http://www-01.ibm.com/software/analytics/spss/products/modeler/index.html. [Accessed 16 11 2012].

[44] "Model Types Supporting PMML," [Online]. Available: http://pic.dhe.ibm.com/infocenter/spssmodl/v15r0m0/index.jsp?topic=%2Fcom.ibm.spss.modeler.help%2Fmodels_pmml_modeltypes.htm. [Accessed 27 08 2012].

[45] SAS Institute Inc., "SAS Enterprise Miner Features," [Online]. Available: http://www.sas.com/technologies/analytics/datamining/miner/index.html#section=3. [Accessed 16 09 2012].

[46] SAS Institute Inc., "SAS/ACCESS Software," [Online]. Available: http://www.sas.com/software/data-management/access/index.html#section=3. [Accessed 16 10 2012].

[47] Rapid-I, "RapidMiner Overview and Features," [Online]. Available: http://rapid-i.com/content/view/181/190/lang,en/. [Accessed 16 11 2012].

[48] "Rapid - I - Operator Overview," [Online]. Available: http://rapid-i.com/content/view/12/34/. [Accessed 18 09 2012].

[49] "Data Mining Algorithms and Tools in Weka," [Online]. Available: http://wiki.pentaho.com/display/DATAMINING/Data+Mining+Algorithms+and+Tools+in+

Weka.

[50] "PMML Support in Weka," [Online]. Available: http://wiki.pentaho.com/display/DATAMINING/PMML+Support+in+Weka.

[51] "KNIME Features," [Online]. Available: http://www.knime.org/features.

[52] "Database Documentation," [Online]. Available: http://tech.knime.org/database-documentation.

[53] "KNIME JavaDoc API," [Online]. Available: http://tech.knime.org/docs/api/.

[54] "Export and Convert R models to PMML within KNIME," [Online]. Available: http://www.knime.org/blog/export-and-convert-r-models-pmml-within-knime.

[55] D. Morent, W.-C. Lin, K. Stathatos and M. R. Berthold, "Comprehensive PMML Preprocessing in KNIME".

[56] M. Lovelace, N. Buchanan, G. Cameron, F. d. Rezende and J. Tarella, "IBM Enterprise Content Management and System Storage Solutions: Working Together," [Online]. Available: http://www.redbooks.ibm.com/abstracts/sg247558.html?Open. [Accessed 09 12 2012].

[57] B. Karwin, "Practical Object Oriented Models In SQL," 22 07 2009. [Online]. Available: http://www.slideshare.net/billkarwin/practical-object-oriented-models-in-sql. [Accessed 18 11 2012].

[58] Rapid-I, "RapidMiner 5.0 User Manual," 2010. [Online]. Available: http://docs.rapid-i.com/files/rapidminer/rapidminer-5.0-manual-english_v1.0.pdf. [Accessed 17 11 2012].

[59] How Products Are Made, "How springs are made," [Online]. Available: http://www.madehow.com/Volume-6/Springs.html. [Accessed 18 10 2012].

[60] G. Svolba, "Efficient “One-Row-per-Subject” Data Mart Construction for Data Mining," in *SUGI 31 Proceedings*, San Francisco, 2006.