# Generate Domain Specific Knowledge to Support Silicon Based MEMS Development

**One Year Master Course Software Engineering**

**S.A.C. Langenhuisen B.Sc.**

1. Thesis Supervisor: Prof. Dr. D.J.N van Eijck
2. Internship Supervisor: Dr. D. Ortloff

Work conducted at: Cavendish Kinetics B.V.
Availability: Public Domain

University of Amsterdam

# Acknowledgements

The last months the student has conducted his master project for the University of Amsterdam at Cavendish Kinetics in ´s Hertogenbosch. Cavendish Kinetics is a spin-off from Cambridge University and was founded in 1994 by Dr. Charles Smith. Cavendish Kinetics focuses on the development of CMOS compatible MEMS process modules, the design and modelling of MEMS devices and subsequently providing these two combined (process module and design module) as an Intellectual Property (IP) package for customers to use in numerous different application areas. The student would like to thank Dr. D. Ortloff of Cavendish Kinetics for his help in forming an interesting and challenging research question; helping to clarify the difficult domain, and for proof-reading several draft versions of this thesis, the project plan, and the literature study. In addition the student would like to thank him for his encouragements to file a patent and publish two papers based on this thesis. The first paper is published in proceedings of the 5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering in Ischia, Italy on the 27th of July 2006 [LO06]. The second paper is published in proceedings of the 11th International Conference on the Commercialization of Micro and Nano Systems in Florida, USA in August 2006 [BVL06]. The student would also like to thank Dr. J. Popp of Cavendish Kinetics for proof-reading this thesis and for his advice in selecting and implementing techniques. His graph-theory knowledge was very useful and helped the student focusing on only a few algorithms. In addition, the student would like to thank B.Sc. B. Veenstra and D. Bor for checking this thesis on English spelling.
From the University of Amsterdam the student would like to thank Prof. Dr. J van Eijck for his effort in proof-reading several draft versions of this thesis, and the time he spent in patiently explaining and formulating difficult equations. In addition, very good and detailed feedback is given where obscurities occurred in draft versions of this thesis.
Finally the student would like to thank his parents, sister, friends, and girlfriend for their support and faith during his study in the past year, but especially during the time spent at Cavendish Kinetics for his master project.

Sven Langenhuisen

´s Hertogenbosch, August 2006

# Abstract

PROMENADE, a sixth framework EU funded project, is to realize a computer-based environment supporting process engineers in creating, verifying, simulating, optimizing and maintaining thin film silicon processes with predictable characteristics. In addition it supports designers of Micro Electro Mechanical Systems (MEMS) devices by offering them a formal interface to constraints from the technological domain and facilitating design for manufacturing.
In the current version of the PROMENADE system, process steps of a new manufacturing process for a MEMS device are verified against a set of consistency rules. These rules capture abstract knowledge about constraints for process steps and flows allowing or disallowing certain combinations or conditions. This rule check allows verification for manufacturability. If all process steps and the whole flow pass this verification stage, a simulation of the process flow will be run to perform the next level of verification. If the MEMS design is verified and successfully simulated, real life experiments are performed. All steps and data generated during this experimental verification is carefully captured and related inside the system.
However, it is still possible that an expensive real life experiment fails. The number of failing experiments in the MEMS development process can be reduced when correlations between process steps, which lead to failing experiments, can be found and abstract knowledge in the form of design rules for the consistency check can be deducted.
This problem is addressed in this thesis by presenting, describing, and validating an approach to generate knowledge, which will be used to (semi) automatically create process step and flow rules for the silicon based MEMS design process. The approach describes a set of concepts and techniques, which are combined and used to generate this domain specific knowledge. Techniques used are ontologies, data warehousing and custom data mining clustering algorithms. The approach is validated within the PROMENADE system.

# Contents

# Glossary

**MEMS** Stands for Micro Electro Mechanical Systems. It is an integration of mechanical structures with microelectronics.

**Process** The fabrication process.

**Process Parameter** Describes constraints of a specific task. E.g.: ”Rotation speed = 5000RPM” or ”Time = 60sec”. For an example, see appendix A.

**Result Parameter** Describes the expected result of a process step. E.g.: ”Thickness = 0.3um”. For an example, see appendix A.

**Process Step** Contains one or more process and/or result parameters, as well as information about up and downside processing. For an example, see appendix A.

**Process Flow** Is a collection of process steps in a specific sequence. A process flow is used to describe the process of manufacturing a specific MEMS device. For more information see appendix A. This appendix is a run card including a complete process flow.

**Run card** Lists the details of the run *(the recipe)*, including process flow, process steps and parameters.

**Data** Defined as raw facts and is unstructured, lacks context and may not be relevant to the recipient. However, data still is the representation of the information.

**Information** Data that is correctly organized, filtered and presented with context it can become information because it then has ”value” to the recipient.

**Knowledge** A pattern in information that is interesting (according to a recipient interest measure) and certain enough (again according to the recipient criteria). The pattern is the basis of what the recipient can do with information.

# 1. Background and Context

## 1.1. PROMENADE

PROMENADE is a process design and tracking system, developed by a consortium of European companies, and is used to verify the approach in this thesis and it allows the specification of processes for specific applications and subsequently the tracking of the development procedures. The PROMENADE system is divided into several environments, which are described in the following sections.

### 1.1.1. Design Environment

The design environment allows management of all data and information relevant to MEMS fabrication process flows and steps. Furthermore, it provides an interface to third party process simulation and visualization tools. The design environment is using models as a vehicle to express how process steps are simulated. These models are generic and powerful since they are the key to flexile simulation and analysis. For example, models can be used to create a cost model for process flows. In addition, the design environment facilitates the work of process designers in a way that it provides a knowledge base consisting of process flows, process steps, parameters, units and materials. Even if the designer is not an expert in every area of MEMS process design, he or she can fall back on a design module, which contains all design related fabrication knowledge of a company [JPB04, DOV05, AWO05, AWO06]. *(See figure 1.1)*.

### 1.1.2. Tracking Environment

Today data and measurement results derived from simulations, tests or measurements during MEMS and deep-submicron process development are usually kept informally and non-systematically on paper, in Excel sheets, or merely in the minds of process engineers. Hence, these research and development results are hardly accessible or re-usable in future process development projects, nor is the development effort reproducible or transparent. The tracking environment is designed to assist and support the process engineer in keeping track of the current developments, as well as structuring the storage and keeping the history of the development efforts. The tracking environment can be used to keep track of all data like projects, experiments, lots, and wafers. But most important is the option to keep track of all artifacts and documents related to the entities mentioned before. For example, it is possible to attach Scanning Electron Microscopy (SEM) images to a wafer entity. [JPB04, DOV05, AWO05, AWO06]. *(See figure 1.1)*.

### 1.1.3. Back Annotation

Back Annotation is an environment, which uses both the design and tracking environment. With this environment it is possible to automatically load data as entities into the PROMENADE system. This thesis describes an approach to generate knowledge and automatically load this knowledge into the design environment. The approach in this thesis will be used in this part of the PROMENADE system [JPB04, DOV05, AWO05, AWO06]. *(See figure 1.1).*

### 1.1.4. Simulation Environment

The last environment is a simulation environment which is used to simulate process flows. Simulation of process flows is the last steps of verification before the flow is send for a real life experiment. The four environments are the PROMENADE system. Figure 1.1 shows the PROMENADE system with its relation to other processes.

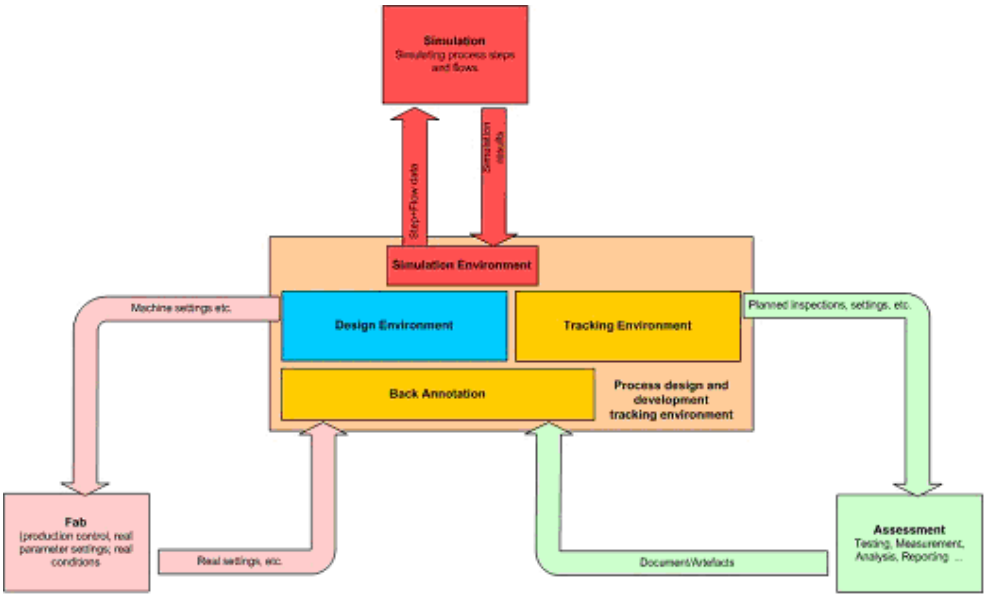


Figure 1.1.: The PROMENADE system

## 1.2. Techniques

In this section a small explanation of the several different concepts and techniques is given, where possible with a reference to used literature.

### 1.2.1. OLAP

Online Analytical Processing (OLAP), or multidimensional analysis, offers the possibility to execute queries that are complex (e.g.: compare sales relative to plan by quarter and region for the prior two years). The results are merely extracted values,

or an aggregation of values [CD97] [Ede96]. OLAP helps to combine data together, but not to create new knowledge. OLAP is a relatively old technique, which has the advantage that the majority of the issues, which always come with new techniques, have been solved [CK05]. Because of the earlier mentioned disadvantages, OLAP will only be used to prepare the information for the actual Data Mining process.

### 1.2.2. Data Warehouse

A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of the data mining process. A data warehouse contains data and delivers it to the data mining tools as information, that can be used to generate knowledge [Erd97]. Typically the data warehouse is maintained separately from the organizations operational databases to prevent having side effects on the performance of the operational database [CD97].

### 1.2.3. Data Mining

Data mining is a technique which reaches much deeper into databases than OLAP. Data mining tools find patterns in the data and infer rules from them. Those patterns and rules can be used to guide decision-making for the creation of new rules for the design environment and forecast the effect of those decisions. This process is represented graphical in figure 1.2. A collection of data and parameters is reduced to a smaller collection of dependencies and patterns. The patterns on their turn are converted into knowledge. The knowledge can be used as guidance to take and forecast decisions. Data Mining is the overall process of examining a data source for implicit information and recording this information in explicit form, in other words, the extraction of high-level knowledge from low-level data [SM98]. In this research project the high-level knowledge is the design rules, generated with the data mining technique. In addition, data mining can analyze data on a very high speed by focusing attention on the most important variables [Ede96]. Low-level data is the data in the PROMENADE system which is loaded into the data warehouse. Data mining can be yielded in five different types: associations, sequences, classifications, clusters, and forecasting. Of these techniques clustering is the most useful technique to use in this context [Kle02, Ede96].

In [WF05, Ede96] a clear view on the several clustering data mining techniques is given. Clustering is a notion that arises naturally in many fields; whenever one has a set of objects, it is natural to seek methods for grouping them together based on an underlying measure of similarity. A standard approach is to represent the collection of objects as a set of abstract points, and define distances among the points to represent similarities. The closer the points, the more similar they are. For this research project different clustering techniques (single-linkage, sum-of-pairs, EM, k-means, and k-median) are checked for suitability.

### 1.2.4. Ontology

Ontology has originated in philosophy as a systematic account on the nature and the organization of reality. The etymology of the word ontology refers to the existence of the world. In the data mining context, which is used for this thesis, ontol-

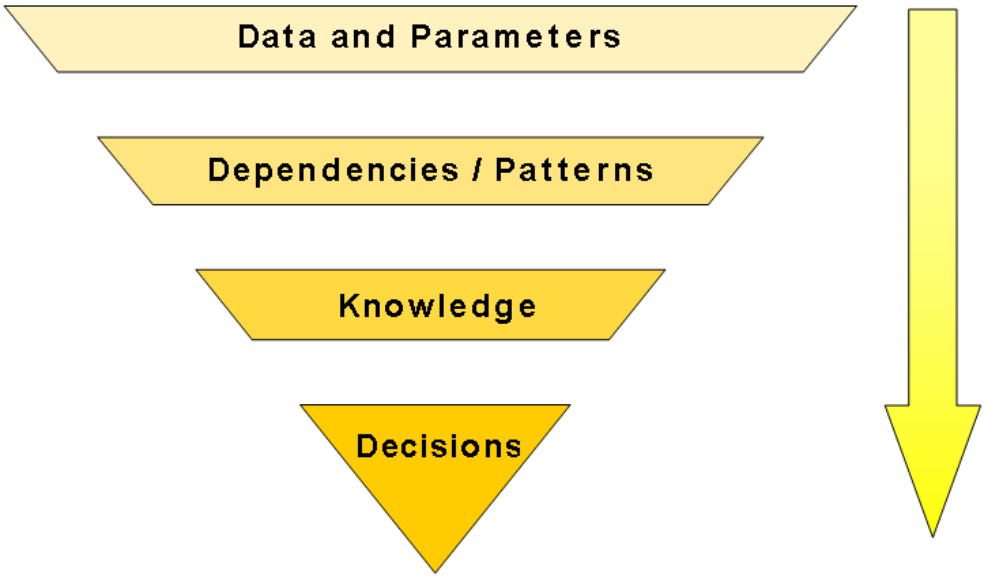


Figure 1.2.: Generate knowledge from data and parameters

ogy is viewed as a formal structure of the system, which encapsulates the semantics of the domain conceptualization. In this sense, the ontology defines the semantics of what is known about the domain that the ontology covers [Gru93, SM98]. A data warehouse, or several different data warehouses [LH04, CD97], can be mapped into an ontology, which makes it easy to query the data warehouse. An ontology can be used in an intelligent knowledge discovery process to increase the rate of success [BP01, SM98].

# 2. Problem Description

The main goal for this research project is to prove that it is possible to generate knowledge for silicon based MEMS development. This chapter describes the current, desired and future situation of the PROMENADE system, upon which the main research question is based, as well as the expected results.

## 2.1. Current situation

The PROMENADE system is divided in several different environments, as can be read in section 1.1. First a user creates process steps and a process flow in the Design Environment (developed by the University of Siegen). When a Process Flow is completed, all individual process steps will be verified against a set of rules (e.g.: aluminum can not be heated warmer than 500 degrees). In addition the whole flow will be checked for consistency.
After verification, the whole process flow will be simulated and if the simulated process flow works correctly, the process flow will be sent to a fab via a run card *(See appendix A)* for a real life experiment *(See figure 2.1)*. At the same time all in-

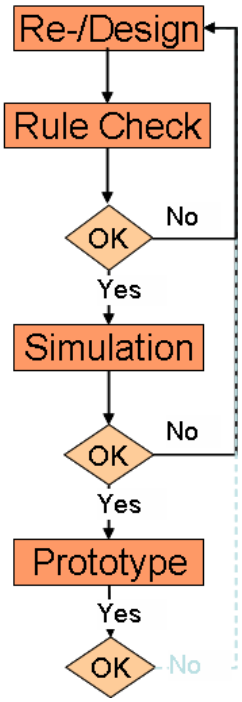


Figure 2.1.: Different stages of verification

formation about the process flow and process steps is sent to the Tracking Environ-

ment via the use of the same run card *(See Figure 2.2)*. Test results of the experiment will be loaded into the Tracking Environment when the test is finished.

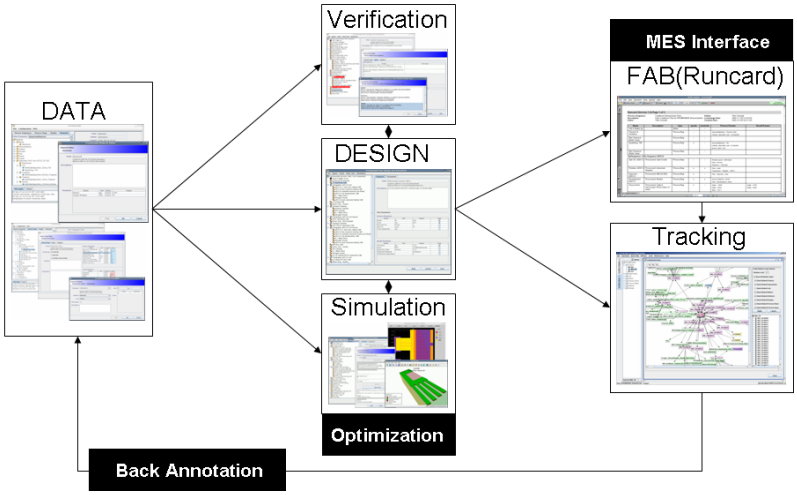


Figure 2.2.: Tracking Environment related to the Design Environment

## 2.2. Desired situation

The main goal at the start of this research project was to find a concept or technique, which can find correlations between process steps, their sequence in a flow ,and test results. (Similar) process steps in a similar sequence in different flows that lead to similar test results need to be related to each other. With these relations it is possible to extract knowledge in the form of design rules for the design environment from the system. Extracting these rules will reduce the number of failing experiments and save both time and money. Techniques that are evaluated for this research project are data mining, OLAP, ontology´s, and the use of a data warehouse.

## 2.3. Future situation

It is necessary to find correlations to (semi) automatically generate rules from the Tracking Environment. In addition these rules need to be delivered to the Design Environment by the Back Annotation environment *(See Figure 2.2)*. Generating rules or presenting graphical user interfaces is not part of the research, but needs to be performed afterwards. This research project is only to prove that knowledge can be generated from data available within the PROMENADE system.

## 2.4. Expected results

The following results are expected at the end of this thesis project.

1. Prototype, to prove that the ideas and algorithms created during this research project work. The prototype has a command line interface and has a clear and simple structure to ensure a smooth integration into the PROMENADE system.
2. Recommendations on how to implement the prototype into the PROMENADE system.
3. Recommendations for future work: *(Semi) Automatic rule generation.*
4. Thesis with a complete and clear explanation of the used approach and the algorithms created and used.

## 2.5. Research questions

The desired and future situations together form the main research goal: *"Is it possible to generate domain specific knowledge to support silicon based MEMS development?"*. Questions which are answered in this thesis are:

1. What is domain specific knowledge for Cavendish Kinetics?
2. Which business intelligence concepts and techniques are available?
   a) How do the different techniques work?
   b) Can a standard approach be used?
3. Is it in the future possible to (semi)automatically generate rules for process steps?

# 3. Research Plan

An approach for the research project, is developed while writing the project plan. This research plan is a detailed approach to answer all research questions, create a working prototype, and is divided into three different phases that together are used to create a complete prototype. This prototype is used to validate the answer to the research question.

## 3.1. Phase 1: Data Mining

A few different steps need to be performed to realize the desired situation mentioned in the previous chapter. The steps are based on the six-step Data Mining Knowledge Discovery process model [JKP98]. This process model provides a complete description of all the steps from problem description to deployment of the results. The steps need to be followed by practitioners when trying to data mine information. The following sections describe the different steps in more detail.

### 3.1.1. Step 1: Understanding the problem

In this step, the problem needs to be defined in cooperation with expert users. Project goals need to be determined as well.

### 3.1.2. Step 2: Understanding the data

The next step in the six-step process model is used to decide what data is needed.

### 3.1.3. Step 3: Preparation of the data

This step is skipped in this research project. The main goal for this research project is to select and validate existing concepts and techniques, or create new techniques to generate knowledge. The data warehouse is filled manually to make sure that enough time is available to perform research on the different data mining clustering techniques. This step is executed later, when the prototype is implemented into the PROMENADE system.

### 3.1.4. Step 4: Data Mining

This is the step of the process model where the actual knowledge is generated. The data mining clustering techniques are used in this approach. Clustering data means that data is analyzed by sorting data into clusters in a way that the degree of association between objects is maximal if they belong to the same cluster and minimal otherwise. This means that process steps form a cluster when their degree of association is high. When process steps have a low degree of association, the process steps do not form a cluster.

Clustering techniques use structures in data to find clusters without explaining why they are clusters. Therefore, it is possible to find clusters of similar process steps, but not tell why this is a cluster. This means that it is not possible to generate new rules without the interference of a process engineer. Data Mining is chosen as the knowledge generation technique based on the literature study. However, a specific suitable data mining clustering technique still needed to be selected at the start of the project. More detailed information about clustering techniques is read first during this research project wherefore suitable papers and books have been selected during the literature study. A technique is suitable when it is possible to match process steps with a covering degree of 100 percent and for example process steps with a covering degree of 80 percent as well. The technique that will be selected needs to be validated against a sample data set and against the data warehouse, which is filled manually. Manually created data is used to validate the data mining step using the *test-first* principle [Shu02].

### 3.1.5. Step 5: Evaluation of the discovered knowledge

In this step the process steps that were found in the previous step need to be presented in an overview to an authorized process engineer. This overview shows the correlation between process steps from different flows that were found, together with the test results of the experiments. In addition, the system need to present an advice for a new rule. The process engineer can adjust this rule before it is submitted into the design environment. According to the project plan the actual generation of new rules and constraints for the design environment is not part of this thesis. An approach to implement this feature into PROMENADE is given in chapter 8.

### 3.1.6. Step 6: Using the discovered knowledge

Newly discovered knowledge will be used automatically as soon as the rule is submitted into the design environment. When a process engineer is using the design environment, he or she can use the newly submitted rule.

## 3.2. Phase 2: Find sequences from clusters

The second phase of the thesis is used to verify if process steps, found in the previous phase, are in similar sequences in process flows. A correlation of a single process step between sequences is not enough. For example, a process step which cleans a wafer occurs most likely in every process flow. Only results of correlations of process steps which occur in a similar sequence in different process flows need to be presented to design engineers. This means that step 5 of the data mining approach needs to be modified as well.

## 3.3. Phase 3: Implement Ontology

The last phase of the research project is used to describe and implement an ontology. The data mining in step four will query the data warehouse via this ontology

which helps to understand the format and size of the data in step two. For example, how is 5 millimeter related to 5 micrometer or 5 nanometer [Gru93]. In this phase an ontology will be implemented to understand and convert data to a uniform format. The other steps of the six-step approach [JKP98] will not be changed.

## 3.4. Combination of techniques

The following set of techniques will be used for this thesis:
*Data Warehouse*
Used to avoid performance problems on the operational databases.
*Ontology*
Used to describe the layout of the data warehouse. An ontology can fulfil step two in the data mining process, "understanding the data" [CK05].
*OLAP*
Step three in the data mining process is "Preparation of the data" and OLAP is used to represent useful data.
*Data mining*
Data mining is the technique that will generate new knowledge, which is going to be used to create new rules for the PROMENADE design environment. Clustering techniques [WF05,Kle02] are the most useful techniques for the problem that needs to be solved. Figure 3.1 shows how the different concepts and techniques will be combined together.

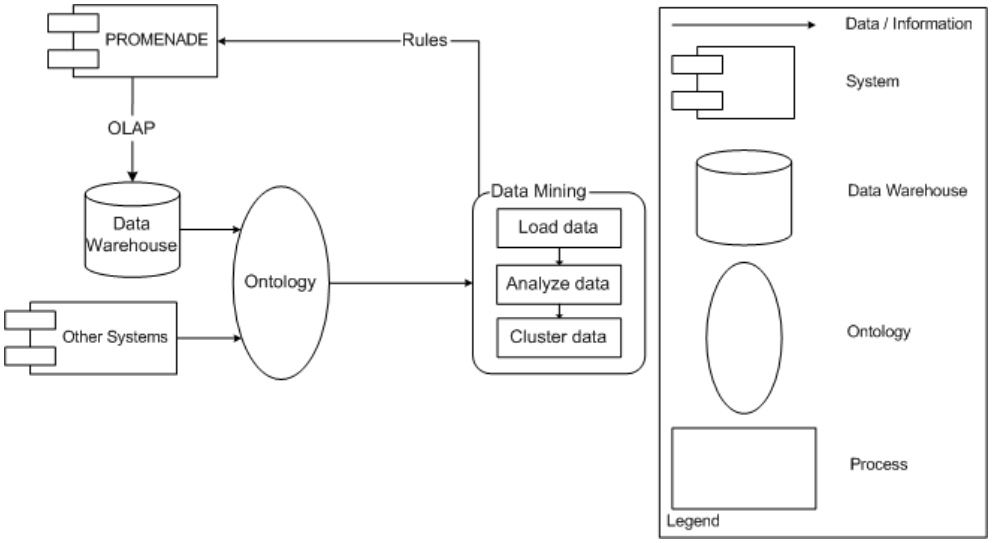


Figure 3.1.: Overview of approach

## 3.5. Future work

To completely implement the prototype, data need to be loaded into the data warehouse. In the future the third step of the six-step process model is automated. Data that is to be used in the data mining process needs to be selected automatically. Normally data has to be cleaned as well. However, as mentioned before, data does not need to be checked for completeness, which automatically means that it does not need to be corrected. All data in the tracking environment can automatically be used for the data mining process. Data stored in the tracking environment is loaded into a warehouse using OLAP. This technique prepares data by combining useful process step data before loading it into the warehouse.

# 4. Plan Execution, Phase 1: Data Mining

This initial approach was developed at the beginning of the research project and describes an approach to find process steps in several different process flows that are similar to each other. The main research question is answered when similar sub-sequences in flows are found. However, in the first phase only correlations between process steps need to be found.

## 4.1. Step 1: Understanding the problem

This thesis describes an approach to (semi) automatically generate knowledge. The actual generation of knowledge is a process that needs to be executed whenever a process engineer wants to. However, this step of the six-step model [JKP98] is executed only once, because the problem and goals are not likely to change. The problem is that experiments are still started with more or less the same process steps in a similar sequence, even when previous experiments failed. Data is automatically loaded from the design environment into the tracking environment as soon as an experiment starts. However, nothing is done with the results of the experiments. The goal of this approach is to present suggestions to users for new rules for the design environment based on results of experiments. With these new rules, the number of failing experiments can be reduced because experiments will not be started at all if it is clear that they are most likely going to fail. This step of the data mining approach is completed during the Literature Study.

## 4.2. Step 2: Understanding the data

As mentioned in the previous section, the kind of data will not change in the future. Just like the previous step of the six-step model [JKP98] this step will be executed only once. A process step will continue to have the same kind of attributes (process parameters and result parameters) and all parameters in a process step will continue to be used as data for the data mining process. All process flows will contain the same attributes (process steps) to form a sequence. This data does not need to be checked for completeness, missing values or plausibility values because all data checks are already in the PROMENADE system. For example, it is not possible to enter a string where a number as a process parameter value is expected.

## 4.3. Step 3: Preparation of the data

As mentioned in section3.1.3 this step of the six-step data mining approach [JKP98] is skipped in this research project. However, this step needs to be performed when this approach is implemented into the PROMENADE system.

## 4.4. Step 4: Data Mining

Data mining itself consists of several different steps, which will be explained using four sample process flows. All flows contain bogus data that is used to demonstrate how data mining is used in this approach.

| *Flow* | *Step* | *Parameter* | *Value* |
|---|---|---|---|
| Flow1 | Step1 | Parameter1 | 15 |
| | Step2 | Parameter2 | 30 |
| | | Parameter3 | 45 |
| | | Parameter4 | 45 |
| Flow2 | Step1 | Parameter1 | 15 |
| Flow3 | Step1 | Parameter1 | 15 |
| | Step2 | Parameter2 | 25 |
| | | Parameter3 | 40 |
| Flow4 | Step1 | Parameter1 | 20 |
| | | Parameter2 | 20 |
| | Step2 | Parameter3 | 45 |
| | | Parameter4 | 10 |

### 4.4.1. Data mining, step 4.1: Place process steps on grid

Every process step needs to be represented as a point on a grid. The definition of a grid for this thesis is a two-dimensional data field in which points can be marked to represent data. Depending on the parameters of each process step a position on the grid needs to be calculated. This is represented in figure 4.1. Calculating the exact position of process steps on a grid is postponed because it was not clear at the start of this phase when two process steps have the same covering degree. The covering degree in the context of process steps is a calculated percentage of how much of a process step is covered by another process step. The calculation of this percentage is explained in section 5.1.1 of this thesis.

### 4.4.2. Data mining, step 4.2: Measure distance between steps

Once all process steps are placed on the grid, the Euclidean distance between two points is calculated. This is probably the most commonly chosen type of distance calculation. It is the geometric distance in the multidimensional space. The formula uses the x and y coordinates of two points, P and Q:

$$P = (Px, Py) \tag{4.1}$$

$$Q = (Qx, Qy) \tag{4.2}$$

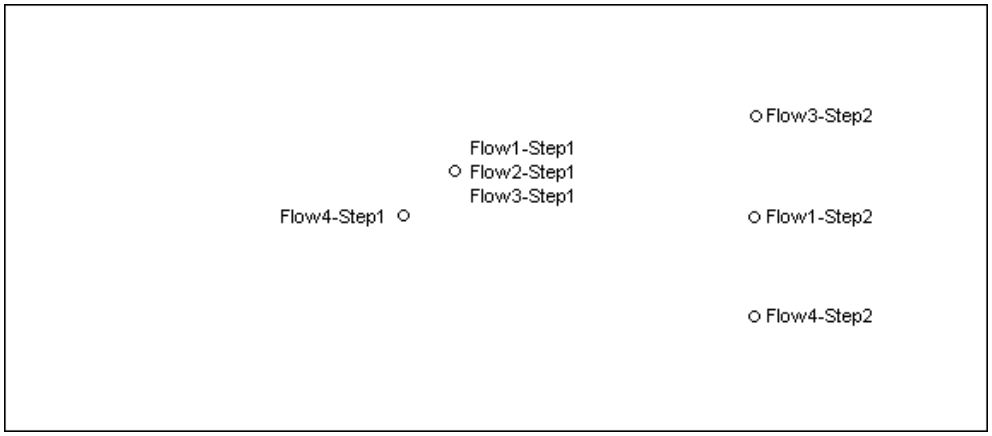


Figure 4.1.: Process steps on the two-dimensional grid

The Euclidean distance between two point is calculated using the following formula:

$$\sqrt{(Px - Qx)^2 + (Py - Qy)^2} \tag{4.3}$$

The smaller the distance between different process steps, the more equal the process steps are. As an example the distance between process steps Flow1-Step1 (f1-1) and Flow2-Step1 (f2-1) will be calculated. First, a mathematical description of both process steps is needed:

$$P(flow1 - step1) = (Px = 100, Py = 105) \tag{4.4}$$

$$Q(flow2 - step1) = (Qx = 125, Qy = 103) \tag{4.5}$$

The distance is measured as:

$$\sqrt{(Px100 - Qx125)^2 + (Py105 - Qy103)^2} \tag{4.6}$$

$$\sqrt{(-25)^2 + (2)^2} \tag{4.7}$$

$$\sqrt{625 + 4} \tag{4.8}$$

$$\sqrt{629} \tag{4.9}$$

$$Distance = 25,079 \tag{4.10}$$

### 4.4.3. Data mining, step 4.3: Linkage

The distance between two process steps is determined in the previous section. All process steps within a specific range of distances are marked as a cluster. A cluster is a collection of process steps that all have a relation with all other process steps in the cluster. The range depends on the accuracy of the correlations between the clusters. This threshold needs to be configurable for the process engineer who is generating knowledge. For example, if the process steps covering degree threshold is set to 35 percent, only a few process steps are marked as a cluster, as can be seen in figure 4.2. In this figure you can see that a process step can be a member of more

than one cluster. This is possible when a process step has a sufficient covering degree by multiple other process steps. But one or more of these process steps do not have a sufficient covering degree by all other process steps in the cluster and are therefore not related to each other. A process step is added to multiple clusters when this situation occurs.

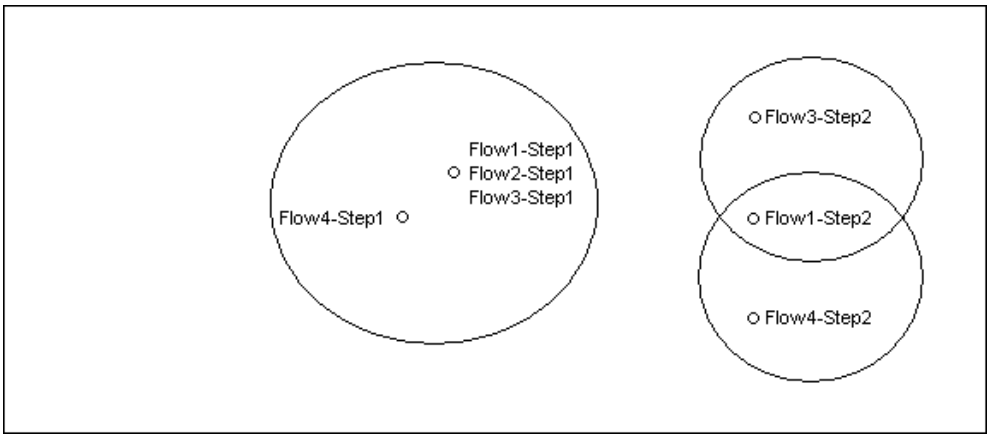


Figure 4.2.: Clustered process steps on the grid

### 4.4.4. Why this approach is not used

A process step contains zero or more process and/or result parameters. This diversity of zero to unlimited parameters makes it very difficult to represent a process step containing so much information as a single point on a two-dimensional grid. Another approach would be to create a three-dimensional grid and again use the distance between points on the grid as a degree of representation used to cluster process steps. But this approach would have the same problem when using a two-dimensional grid. The disadvantage still is that it is hard to find an approach to calculate positions of process steps on a grid, which needs to be performed manually before coding the prototype because the *test-first* principle is used. The advantage of this approach is that it is easy to detect which process steps form a cluster, because these process steps are all marked nearby each others on the grid.
However, the disadvantages which maybe were not solvable within the timeframe of this research project forced to use another approach which is presented in the next chapter.

# 5. Plan Execution, Phase 1: New approach

The approach in the previous chapter describes how distances between process steps on a grid can be used to find clusters of process steps. However, as mentioned in the previous chapter, it is very difficult to implement this approach. Therefore, an other approach is developed, which uses relations and graphs to find correlations based on the degree of representation between process steps. A uni-directional relation between two process steps is created when a process step X is covered enough by another process step Y. This relation becomes bi-directional, or symmetric, when process step Y is covered enough by process step X. Why the relation between process steps is not automatically symmetric is explained further in this chapter. The relations between process steps can be used to find the biggest cluster of process steps where all relations in a cluster are reflexive, transitive, and symmetric. This means that all relations are equivalent [TR01]. This approach is better than the previous approach because it now is possible to manually calculate the covering degree between process steps and manually draw a graph. This is better because now the results of the algorithms can be compared to the results calculated manually in advance. The graph will be created in advance to use as a tool to validate the approach. Another advantage of this approach of creating a graph is that it now is possible to adapt standard graph-theory algorithms. More detailed information can be found in the following sections.

## 5.1. Step 4: Data Mining

### 5.1.1. Data mining, step 4.1: Pre-Clustering of Process Steps

The first step is comparing a process step X with all other process steps and checks if the other process step covers process step X enough. The covering degree is only verified into a single direction e.g.: from process step X to Y. A uni-directional relation between process steps X and Y is created when process step X is covered enough by process step Y. This relation is not bi-directional because the algorithm is not symmetric, which is explained further in this section. Only if process step Y is covered enough by process step X, the relation will become bi-directional.
The covering degree is measured by the deviation of a comparing process step Y to the compared process step X. The deviation of a single parameter has to be less than the value of the original parameter value. E.g.: if a parameter has a value of 5.0, the comparing parameter can have a maximum value of 9.99. If the deviation is equal or bigger than the original parameter value, the degree of representation is 0 because the deviation is too large to be used for this domain. This fact means that the algorithm is not symmetric. For example, take two process steps (X and Y) both containing the same parameter, one with value 5(X) and one with value 10(Y).

The covering degree from process step X to Y will be 0; this is because the value of the comparing parameter from process step Y is twice the size of the original parameter from process step X. The covering degree of process step Y to X will be 0,5 because the value of the parameter from process step X is half the value of the comparing process step Y.
As mentioned before in this section, the approach compares a process step X with all other process steps. For example, take two process steps (X and Y) both containing the same parameter, one with value 3(X) and one with value 4(Y). If process step X is used to measure the degree it is covered by process step Y; the result will be 0,66 if the implementation of the function *ParDif*, which is explained further on in this thesis, is used. If the same function is used to measure the covering degree of process step Y by process step X the result will be 0,75.
This implementation is not symmetric because the deviation of a process step X with respect to process step Y has other proportions than the deviation of the same process step Y with respect to process step X.
A few different steps are performed in this pre-clustering step to determine if a process step is covered enough by another process step:

1. Determine the covering degree (of process step A by process step B) of parameters that occur in both process steps
    a) Select all parameters that occur in both process steps *(intersection)*
    b) Determine the covering degree of all parameters that occur in both process steps
    c) Sum all covering degrees together and divide this by the number of parameters that occur in both process steps
2. Measure how many parameters of a process step A occur in a comparing process step B
3. Measure how many parameters of a comparing process Step B occur in a process step A

The formal specification for this pre-clustering step is:

1. Determine the percentage of parameters from StepX that is present in StepY (5.2)
2. Determine the percentage of parameters from StepY that is present in StepX (5.3)
3. Determine the covering degree of the values from the parameters of StepX and StepY (5.4). The sum of all covering degrees is divided by the number of parameters occurring in both process steps to get the covering degree of all parameters.
4. The results of (5.2) times (5.3) times (5.4) must be bigger than P. This range needs to be changeable in the final implementation of the prototype. (5.5)

$$Rresult = \{(Stepx, Stepy) | Stepx, Stepy \in STEP, \tag{5.1}$$

$$\frac{CommonPar}{ParName(Stepx)} \times \tag{5.2}$$

$$\frac{CommonPar}{ParName(Stepy)} \times \tag{5.3}$$

$$\frac{\sum(ParDif(ParVal(x), Parval(y)) | x, y \in CommonPar)}{CommonPar} \tag{5.4}$$

$$\geq P\} \tag{5.5}$$

CommonPar : $ParName(StepX) \cap ParName(StepY)$
ParName : Function which returns a set of parameter names
ParDif : Abstract function which returns the percentage of value covering between process steps
ParVal : Function which returns a value of a parameter
STEP : Set of all steps

The function *"ParDif"* is an abstract function that can be changed if desired. The function for this thesis is implemented as an non-symmetric function, which uses positive parameter values that are not equal to zero as input, and returns the covering degree as a number between 0 and 1. If needed, the function can be rewritten to be symmetric or adopt other parameter types. E.g.: Different materials that on a similar position in the hierarchy can be related together. The formal specification of the implementation of the abstract function *"ParDif"* used for this thesis is:

$$ParDif = \begin{cases} 0 & \frac{ParValueY}{ParValueX} \geqq 2 \\ 2 - \frac{ParValueY}{ParValueX} & \frac{ParValueY}{ParValueX} \geqq 1 \\ \frac{ParValueY}{ParValueX} & otherwise \end{cases} \tag{5.6}$$

To illustrate the previous equations, two smaller examples will be explained.

Example 1:
In the following example the values of the parameters are not similar:
The parameters do not have the same value, so the result of equation 5.4 will not

| *Parameter* | *Value* |
|---|---|
| Step 1 ParameterA | 5 |
| Step 1 ParameterB | 5 |
| Step 1 ParameterC | 5 |

| *Parameter* | *Value* |
|---|---|
| Step 2 ParameterA | 10 |
| Step 2 ParameterB | 10 |

be symmetric. For the relation process step 1 to process step 2 this will be: ((5/10) + (5/10)) / 2 = 1/2. The number of parameters from process step 1 that occur in process step 2 is two out of three. The number of parameters from process step 2 that occur in process step 1 is two out of two. Process step 2 covers process step 1 for (0.50 times 0.66 times 1 =) 0,33 times 100 = 33% . The other way around the result is different: ((10/5) + (10/5)) /2 = 2/2. However, according to the equation, the similarity of parameters is 0 when the result is bigger or equivalent to 2. So the result is: (0 times 1 times 0.66 =) 0,00 times 100 = 0%. Since the results of the calculations are different, the relation is a-symmetric.

Example 2:
All parameters that occur in both process steps have the same value, so the value

| *Parameter* | *Value* |
|---|---|
| Step 1 ParameterA | 10 |
| Step 1 ParameterB | 10 |
| Step 1 ParameterC | 10 |

| *Parameter* | *Value* |
|---|---|
| Step 2 ParameterA | 10 |
| Step 2 ParameterB | 10 |

of equation 5.4 will be 1. The number of parameters from process step 1, which occur in process step 2 is two out of three. The number of parameters from process step 2 which occur in process step 1 is two out of two. Process step 2 covers process step 1 for (1 times 0.66 times 1 =) 0,66 times 100 = 66% . The other way around the result will be (1 times 1 times 0.66 =) 0,66 times 100 = 66%. This relation is symmetric due to the fact that all values of parameters occurring in the both process step are completely similar.

The formal specification is translated into a Java algorithm, which makes it usable in the PROMENADE system. The test data from the table in section 4.4 is used to initially validate the algorithm. A validation using a bigger data set has been carried out as well; however the results are not mentioned in this thesis due to the size of the data and resultset. More information about this test can be found in 8.3.2. The results of the pre-clustering step are not symmetric and mentioned in the following table and figure 5.1 where the minimum covering degree is 35 percent:

| *Source* | *Target* | *DegreeOfRepresentation* |
|---|---|---|
| Flow1-Step1 | Flow2-Step1 | 100.00% |
| Flow1-Step1 | Flow3-Step1 | 100.00% |
| Flow1-Step2 | Flow3-Step2 | 57.41% |
| Flow1-Step2 | Flow4-Step2 | 40.74% |
| Flow2-Step1 | Flow1-Step1 | 100.00% |
| Flow2-Step1 | Flow3-Step1 | 100.00% |
| Flow3-Step1 | Flow1-Step1 | 100.00% |
| Flow3-Step1 | Flow2-Step1 | 100.00% |
| Flow3-Step2 | Flow1-Step2 | 55,83% |
| Flow4-Step1 | Flow1-Step1 | 37.50% |
| Flow4-Step1 | Flow2-Step1 | 37.50% |
| Flow4-Step1 | Flow3-Step1 | 37.50% |

### 5.1.2. Data Mining, step 4.2: Clustering of Process Steps

The result of the pre-clustering step will be used for the final clustering step, in which the biggest possible clusters will be created.
The algorithm will take a node in the graph and will try to add all related nodes to a cluster where all nodes have a transitive and symmetric relation with all other nodes in the cluster. An additional cluster is created when a node does not have a transitive and symmetric relation with all other nodes in the original cluster. This always guarantees that the biggest possible clusters are detected.

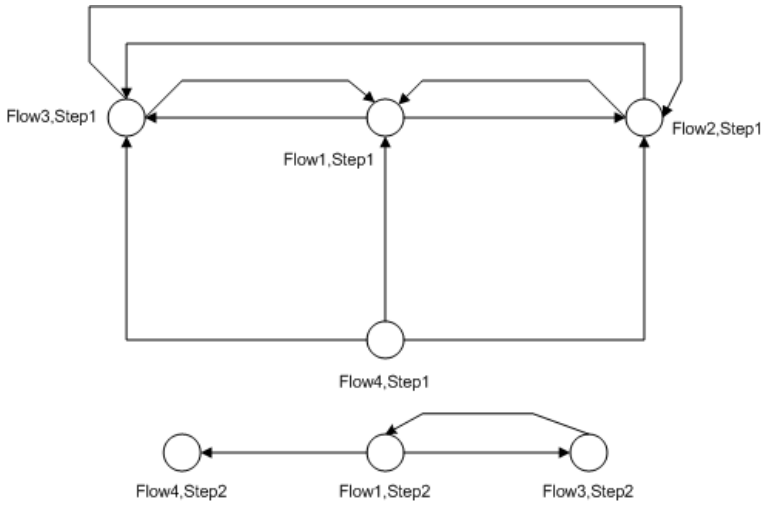


Figure 5.1.: Relations between process steps

The formula presented in section 5.1.1 is not symmetric. This means that a relation from process step X to process step Y does not imply that a relation from process step Y to process step X exists. First all a-symmetric relations need to be removed from the result list of the previous step. This is performed with equations 5.7 to 5.9.

$$Rresult = \{(StepX, StepY) | \forall StepX, StepY \in Rpreclustering, \tag{5.7}$$

$$StepX \rightarrow StepY, \tag{5.8}$$

$$StepY \rightarrow StepX\} \tag{5.9}$$

Rpreclustering : Result of previous step

Equations 5.10 to 5.13 create the biggest possible subset of a graph where all relations are symmetric, reflexive, and transitive. First, equation 5.11 verifies that all nodes in the subset have relations to all other nodes in the cluster. Equation 5.12 creates the biggest possible clusters of nodes with a minimal size of 2 and a maximum size of N. This is the case in the next example given in figure 5.2. However,

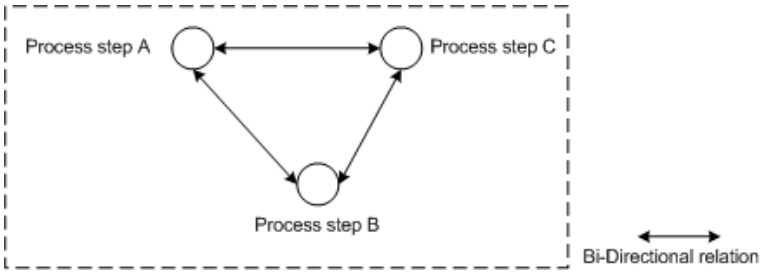


Figure 5.2.: Example: One cluster

when a process step, and the relations between the new and current process steps in the cluster, do not meet all conditions, a new cluster is created, as can be seen in figure 5.3.

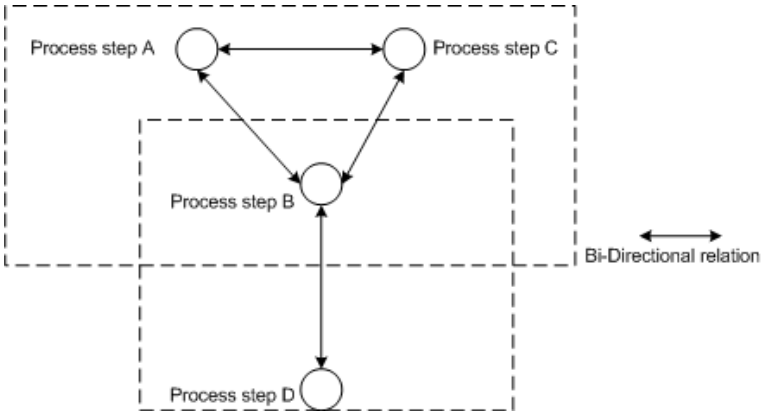


Figure 5.3.: Example: Two clusters

The formal specification of this clustering step is:

$$Cresult = \{(Step1....StepN)| \tag{5.10}$$

$$(StepX \rightarrow StepY) \in X \Leftrightarrow \forall x, y, \tag{5.11}$$

$$x \neq y \wedge 1 \leqq x, y \leqq N, \tag{5.12}$$

$$(StepX, StepY) \in Rresult \tag{5.13}$$

Rresult : Result of previous step

The equation above solves a traditional problem, the clique problem [Mis06a, TR01]. A clique is a complete sub graph of the original graph. The maximum clique problem is the optimization problem of finding a clique of maximum size in a graph. A brute force algorithm to find a clique in a graph is to examine each sub-graph and check to see if it forms a clique. The brute force approach has a disadvantage that it has a "non-deterministic polynomial time". This means that each different combinations of process steps that can be "verified" by a deterministic Turing machine are checked in polynomial time which means that there is no faster solution to find answers. The speed of the algorithm is: $O(n^n)$

Like the pre-clustering formula, the clustering formula is translated into a Java algorithm to make it usable in the PROMENADE system. The test data from the table in section 4.4 is used to initially validate the algorithm. Again, the algorithm is validated using a bigger data as well. The results of the clustering step are marked in figure 5.4 with a dotted line.

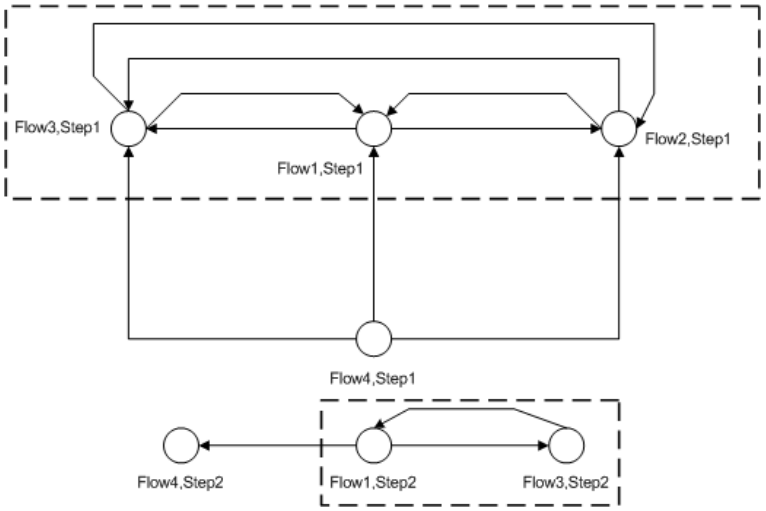


Figure 5.4.: Clustered process steps

# 6. Plan Execution, Phase 2: Find correlations between sequences

## 6.1. Step 4: Data Mining

The first three steps of the six-step data mining approach [JKP98] are completed during the previous phase. Step four could be marked as completed as well because process steps are mined in the data warehouse. However, the knowledge gained in this step is neglectable. It is nice to know when different process flows contain comparing process steps, but knowledge becomes valuable when process steps in different flows in a similar sequence can be detected. Our previous test dataset in 4.4 was not prepared for this extension of the data mining process. The previous test dataset did not contain any information about the position of a process step in a flow. Therefore the test data set is extended with information about the position of a step in a flow. Therefore the data warehouse data model is changed as well.
The values of parameters in the new test dataset are all similar, which assures that an isolated part of the algorithm is tested. However, variable parameter values are used while validating against a bigger data set. The updated test dataset is:

| *Flow* | *Step* | *Position* | *Parameter* | *Value* |
|---|---|---|---|---|
| Flow1 | Step1 | 1 | Parameter1 | 10 |
| | Step8 | 2 | Parameter1 | 10 |
| | Step12 | 8 | Parameter1 | 10 |
| Flow2 | Step1 | 1 | Parameter1 | 10 |
| | Step8 | 2 | Parameter1 | 10 |
| | Step21 | 3 | Parameter1 | 10 |
| | Step12 | 6 | Parameter1 | 10 |
| Flow3 | Step1 | 3 | Parameter1 | 10 |
| | Step8 | 4 | Parameter1 | 10 |
| | Step21 | 5 | Parameter1 | 10 |

### 6.1.1. Data mining, step 4.3: Create sub-sequences

The output of the algorithms using the test dataset from section 6.1 will be a collection of clusters that each contain similar process steps. As mentioned before, important knowledge is gained when correlations of process steps sequences in different process flows can be found. Therefore, sequences need to be created based on the clusters from the previous phase.

$$Ssequence = \{(Step1....StepN)|StepX, StepY \in Cresult, \quad (6.1)$$
$$StepXflowName = StepYflowName, \quad (6.2)$$
$$\exists Step[distance(StepX, StepY) = 1]\} \quad (6.3)$$

distance : Function which calculates distance between process steps in a flow
Cresult : Result of previous step

Equations 6.1 to 6.3 will create sequences of all process steps that are in the same process flow. An extra condition (6.3) is that the distance between the position of a process step in the process flow and the position of any other process steps in the process flow can not be larger than one. This means that the whole sub-sequence is interconnected. A new sequence is created when the distance is greater than one. Figure 6.1 shows an example process flow. The distance between process


Figure 6.1.: Use distance to create subsequences

step 1 and process step 2 is one, so these steps form a sub-sequence. The distance between process step 2 and process step 8 is five. However, the maximum distance between process steps is one, as can be seen in equation 6.3. The distance between process step 8 and process step 9 is one again. The result of this equation will be two different subsequences, namely: ”Step1-Step2” and ”Step8-Step9”.

### 6.1.2. Data mining, step 4.4: Pre-Clustering of Process Flows

The next step in the data mining process is to relate similar subsequences of process flows together, just like relating individual process steps together in section 5.1.1. However, relating process flows together is easier to accomplish than relating process steps. Two sub-sequences of process flows are similar when both contain the same process steps. This simplified version of data mining step 4.1 of the data mining process ensures bi-directional relations, which implies that the algorithm in this step is symmetric.

$$Rresult = \{(SequenceX, SequenceY)| \quad (6.4)$$
$$SequenceX, SequenceY \in Ssequence, \quad (6.5)$$
$$SequenceX \iff SequenceY\} \quad (6.6)$$

Ssequence : Result of previous step

Like the previous steps, the specification is translated into a Java algorithm, which makes it usable in the PROMENADE system. This time new test data is used which is represented in section 6.1. The results of the pre-clustering step are marked in the following table and figure 6.2

| *Source* | *Target* |
|---|---|
| Flow1: Step1 - Step8 | Flow2: Step1 - Step8 |
| Flow1: Step1 - Step8 | Flow3: Step1 - Step8 |
| Flow2: Step1 - Step8 | Flow1: Step1 - Step8 |
| Flow2: Step1 - Step8 | Flow3: Step1 - Step8 |
| Flow3: Step1 - Step8 | Flow1: Step1 - Step8 |
| Flow3: Step1 - Step8 | Flow1: Step1 - Step8 |
| Flow2: Step1 - Step8 - Step21 | Flow3: Step1 - Step8 - Step21 |
| Flow3: Step1 - Step8 - Step21 | Flow2: Step1 - Step8 - Step21 |
| Flow2: Step8 - Step21 | Flow3: Step8 - Step21 |
| Flow3: Step8 - Step21 | Flow2:Step8 - Step21 |

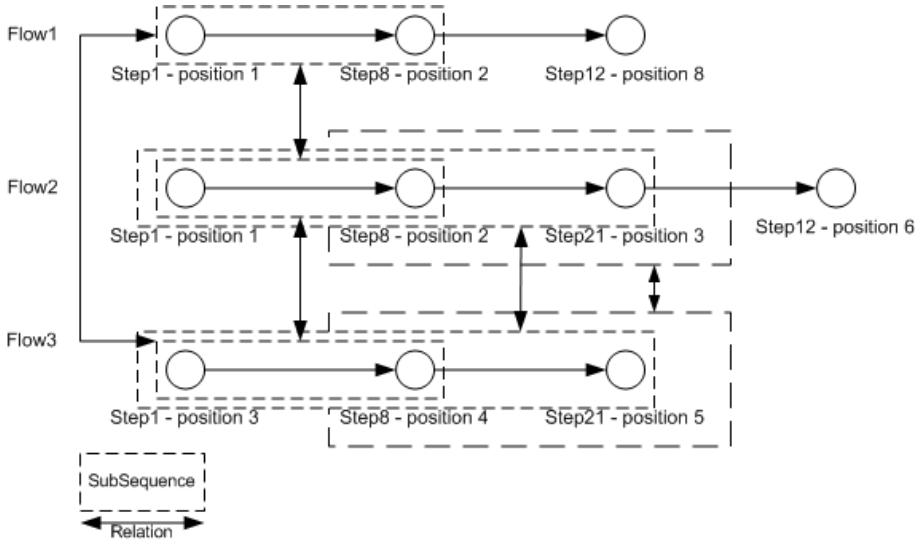


Figure 6.2.: Relations between (sub)sequences

### 6.1.3. Data mining, step 4.5: Clustering of Process Flows

The final step in this data mining process is to find the biggest possible clusters. This is a traditional problem, like clustering process steps in section 5.1.2. Therefore the formal specification for this clustering step is similar to the formal specification used for clustering process steps with some minor updates.

$$Cresult = \{(Sequence1....SequenceN)| \tag{6.7}$$

$$(SequenceX \rightarrow SequenceY) \in \mathbb{X} \Leftrightarrow \forall x, y, \tag{6.8}$$

$$x \neq y \wedge 1 \leqq x, y \leqq N, \tag{6.9}$$

$$(SequenceX, SequenceY) \in Rresult \tag{6.10}$$

Rresult : Result of previous step

Again, this equation tries to solve the clique problem [Mis06a,TR01]. As mentioned in section 5.1.2 this approach has a "non-deterministic polynomial time"(NP time). Process step clustering has a NP time as well. This means the time that will be

consumed by this step of the data mining process will increase exponentially even further ($O(n^n)$). An advice to keep the time for the data mining process limited is given in chapter 8.

The clustering formula is translated into a Java algorithm to make it usable in the PROMENADE system and to complete the prototype. Again, the test data from section 6.1 is used to initially validate the algorithm followed by a validation using a bigger data set. The results of the clustering step are marked in figure 6.3 with a dotted line:

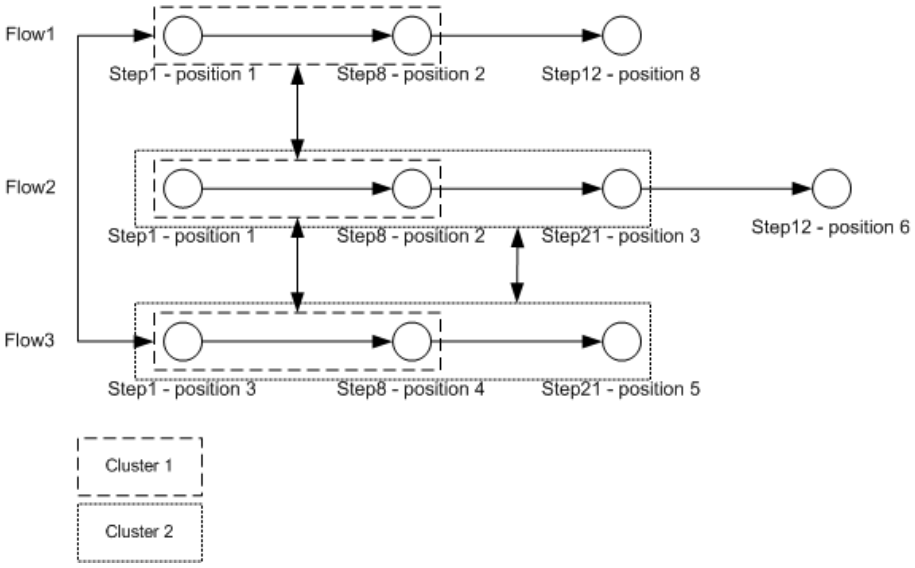


Figure 6.3.: Clustered (sub)sequences

# 7. Plan Execution, Phase 3: Implement Ontology

The final phase of this research project is to implement an ontology. The main goal of this phase is to find an approach to compare different units used in process parameters, for example millimeters with micrometers or seconds with minutes. The original plan was to use an ontology to query the data warehouse with data from the PROMENADE system and other miscellaneous systems at the same time.
While using the PROMENADE system to validate the previous phases it became clear that the conversion functionality is already available. The design environment of the PROMENADE system uses a module called PUMA (Parameters, Units and Materials) which is capable to convert units into other units. For example, in figure 7.1 the unit millimeter is converted into nanometer. Therefore it is decided

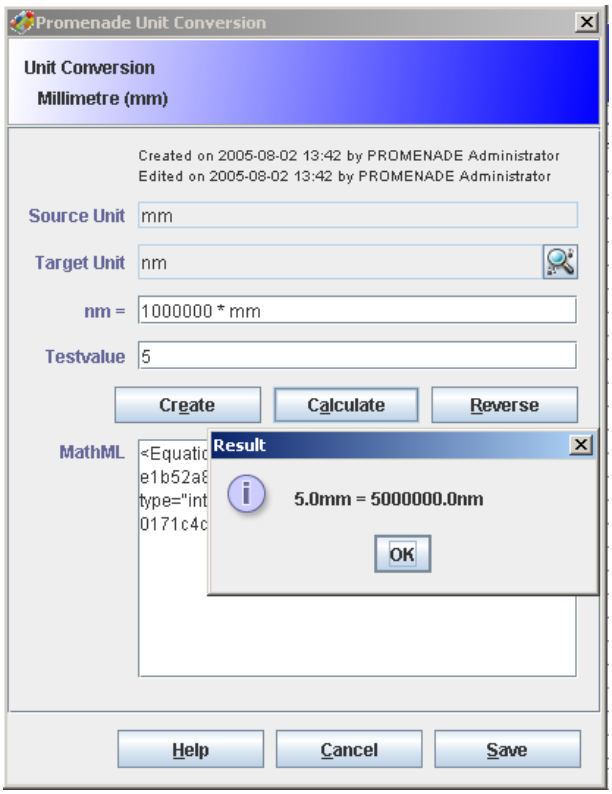


Figure 7.1.: Unit conversion in PROMENADE

not to implement a complete new ontology but to propose an approach to use all currently available functionalities from the PROMENADE system. As mentioned in section 3.5, no tool will be written to automatically load data from the PROME-

NADE system into the data warehouse used for data mining. When the approach presented in this thesis is implemented into the PROMENADE system, all process parameters from process steps need to be converted using similar units. This can be achieved to convert all similar unit types to a similar format, for example convert all micrometers, nanometers etc. to millimeters and all milliseconds and minutes etc. to seconds.

Another functionality of the PROMENADE system is the import/export function using Process Development Tracking Markup Language(PDTML). It is possible to load new data in the PROMENADE system via this predefined format. Since this functionality is already available, it is not necessary to implement an ontology to map different systems. A disadvantage of implementing a new ontology would be that data needs to be checked. The PROMENADE system verifies all data, even when process engineers manually change data; however, other systems, mapped into the ontology, might allow invalid data. Therefore it is decided to use the PDTML import functionality that ensures data completeness, cleanness, and validation. Figure 7.2 shows how the different concepts and techniques are combined together when the prototype is completely implemented into the PROMENADE system.

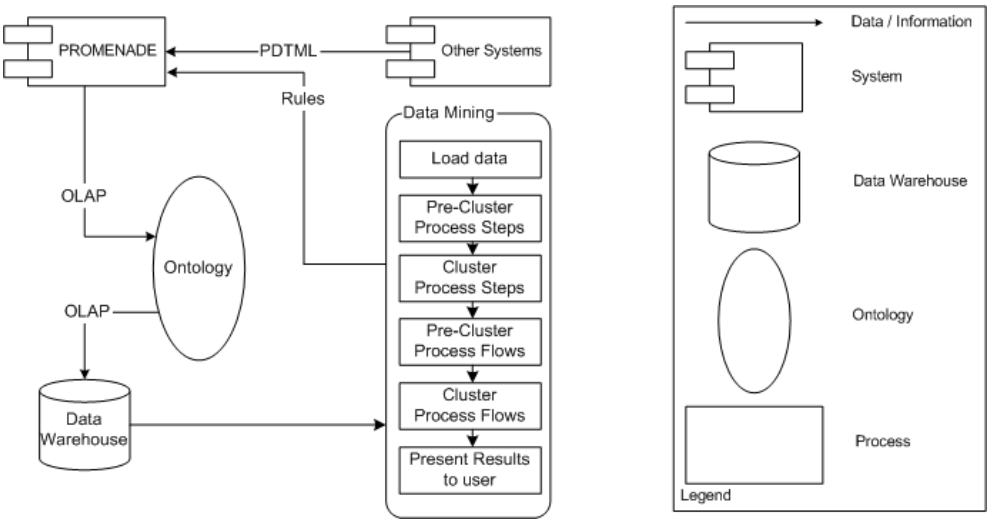


Figure 7.2.: Overview of prototype (when implemented)

# 8. Results

## 8.1. Achieved results

The approach proposed in this thesis is based on the six-step data mining approach [JKP98]. For all steps of the approach a specific interpretation had to be given, where the data mining step (step 4) contained too much information for a single step. Therefore an extra deviation was made:

1. Pre-clustering process steps
2. Clustering process steps
3. Parse sequences
4. Pre-clustering sequences
5. Clustering sequences

Step two and five of this extra deviation use adapted standard approaches to solve graph theory questions [TR01]. Step one, three and four are supporting steps to relate process steps and sequences of process flows together and create a graph of the relations. More detailed information like a formal specification can be found in chapters 4,5, and 6.
The results that are mentioned in all chapters were validated by hand. Except for one big test dataset, all datasets were small enough to calculate the result of the algorithms by hand. These results were calculated before the formal specification was implemented into a Java algorithm, which guaranteed that the algorithms were written to match the results of the algorithm and the manually calculated results. This approach is based on the *test-first* principle where code is written to pass all tests, instead of writing tests, where results always match with the results from algorithms, when coding is finished [Shu02].
Aside from validating the algorithm on a small dataset, a test using a big dataset was carried out as well that contained as much process flows and process steps as developed by Cavendish Kinetics for the last three years. First a number of unique process flows and process steps were added, followed by a number of manually calculated similar process flows and steps. Therefore even results of the big dataset could be predicted and verified. A summary of test results can be found in section 8.3.2. All final results of all tests are correct and verified by the manual calculations.

## 8.2. Similar problems

The PROMENADE system is a rather innovative product and only a few competitors are known that have products offering similar functionality. However, none of these products support functionalities to generate knowledge for the silicon based

MEMS development, and especially for finding correlations between process steps and their sequence in a flow. This approach is completely new and therefore a patent is filed in by Cavendish Kinetics.
On the algorithm level a few adapted standard approaches are used. Process steps in the PROMENADE system are custom data objects, which are hard to correlate together due to the variety of parameters in numbers and values. Therefore, it was not possible to use standard algorithms to correlate the process steps. However, by representing the relations between process steps and sequences of process flows in a graph, it became possible to adapt standard approaches to solve problems, like finding a maximum clique in the graph [JKP98]. Given the fact that the results, as shown in the previous section, are as expected and a patent is filed in for the idea and approach, the approach presented in this thesis can be considered as successful.

## 8.3. Implementation

The approach presented in this thesis can be implemented in the PROMENADE system without any problems. The prototype contains a *connector* class to connect it to a data warehouse, which is a MySQL database for the prototype. However, this *connector* class can be modified to use any other data source. It is advised to create an export tool for the PROMENADE system that loads all data from the PROMENADE system into a MySQL database, the warehouse, which can be queried by the code used in this prototype. As mentioned in section 1.2.1, OLAP as a technique will help loading data in the warehouse. OLAP will combine data to make it useful for this data mining approach. The advice to use a separate data warehouse is given to prevent side effects on the performance of the operational database. However, if desired, the *connector* class can be modified to use the operational Oracle/PostgreSQL database. The current data model of the PROMENADE system does not support position numbers of process steps within a process flow. However, this is a minor addition that is going to be implemented when the data model is updated. Position numbers of process steps need to be calculated while filling the data warehouse, if the prototype is implemented before the data model of the PROMENADE system is updated.

### 8.3.1. Create UI

The prototype is using a plain and simple command line interface. The focus of the research project is to prove the concept of generating knowledge, not creating a completely usable user interface. An interface needs to be created that enables process engineers, who are going to use the tool, to modify settings that are now set in the command line.
Besides the settings, also the results of all algorithms need to be visually presented to the process engineer. The results of all steps using the test data from section 6.1, gives the graph, mentioned in figure 6.3, as a result.

The results of this prototype are presented like the following lines, which contains all data, but in a unusable interface.

```
SHOWCluster: 0
Node 1: flow2
Process Steps in sequence: 3
Step: flow2-Process Step 1 at position 1 with ID 2
Step: flow2-Process Step 8 at position 2 with ID 4
Step: flow2-Process Step 21 at position 3 with ID 6
Node 2: flow3
Process Steps in sequence: 3
Step: flow3-Process Step 1 at position 3 with ID 3
Step: flow3-Process Step 8 at position 4 with ID 5
Step: flow3-Process Step 21 at position 5 with ID 7
SHOWCluster: 1
Node 1: flow1
Process Steps in sequence: 2
Step: flow1-Process Step 1 at position 1 with ID 1
Step: flow1-Process Step 8 at position 2 with ID 10
Node 2: flow3
Process Steps in sequence: 2
Step: flow3-Process Step 1 at position 3 with ID 3
Step: flow3-Process Step 8 at position 4 with ID 5
Node 3: flow2
Process Steps in sequence: 2
Step: flow2-Process Step 1 at position 1 with ID 2
Step: flow2-Process Step 8 at position 2 with ID 4
```

### 8.3.2. Speed issue

The only downside of the approach used, is the speed of the algorithms. The actual clustering steps of the approach have a non-deterministic polynomial time, as mentioned before. The disadvantage is that the algorithm is slower when more item paths in a graph need to be verified. This means that if more process steps and flows are stored in the data warehouse, the algorithm will get slower. The following tables show the results of measurements while running the prototype on a local workstation, comparable with a workstation of process engineers, and the results of measurements running the prototype on the application server. The test dataset used contains 320 process flows with an average of 48 process steps each containing 3 or 4 process parameters. The total number of process steps is 15300 and the total number of process parameters is 45950. All measurements are only executed twice and therefore can only be used as an indication of the speed.
Evaluation of speed results on a workstation:

| *SimilarProcessSteps* | *CalculatedClusters* | *AlgorithmClusters* | *Time* |
|---|---|---|---|
| Minimal 25 | 1 | 1 | 51 sec. |
| Minimal 5 | 2 | 2 | 581 sec. |
| Minimal 2 | 2 | 2 | 946 sec. |

Evaluation of speed results on the application server:

| *SimilarProcessSteps* | *CalculatedClusters* | *AlgorithmClusters* | *Time* |
|---|---|---|---|
| Minimal 25 | 1 | 1 | 72 sec. |
| Minimal 5 | 2 | 2 | 341 sec. |
| Minimal 2 | 2 | 2 | 523 sec. |

As can be seen in the results, the prototype is much faster when the minimum length of correlations between sequences of process flows is increased. With increasing the minimum length, the number of sub-sequences is decreased, resulting in an increase of speed. The application server is slower in comparing flows with a minimal similarity of at least twenty-five process steps because the Java Hotspot client starts compiling lines of code when executed 10.000 times, instead of 1.500 times at the workstation. More information on the Java HotSpot client can be found in [Mic02]. An advice to reduce the speed issue is given in the next section.

## 8.4. Future research directions

Although more time is not available, the research is not completely finished. As mentioned before in this chapter, the clique problem [Mis06a,TR01] is solved using a brute force method. However, this method has a non-deterministic polynomial time. To prevent this more research needs to be done in heuristic methods to resolve the clique problem [TR01]. Not all nodes in a graph need to be compared, if it is possible to predict where in the graph a maximum clique can be found. In [Mis06a] an algorithm called *union-find* is proposed, however in [TR01] no real solution is proposed. It is claimed that this problem is NP-hard, and as such, it is considered unlikely that there exists an efficient algorithm for solving it. Therefore, an approach to find a proper solution, based on heuristics, will have to start with a new literature study especially for this problem. Another approach would be to adapt standard algorithms for the strongly connected components [TR01, Mis06b]. Although the graph created in the pre-clustering steps is not a directed graph, maybe this approach can still be useful.
Another new research direction would be to find correlations between sequences in process flows where for example two process steps are similar, one process step is different, and the next two process steps are similar again. At this moment only process steps in a complete sequence are related together. An approach to complete this would be to find standard algorithms to compare sequences. In [TR01] such kind of approaches and algorithms can be found.

# 9. Evaluation

Although the algorithm is suffering from a speed issue, the requirements set before the research project started are fulfilled. This approach proves that it is possible to generate new knowledge for the silicon based MEMS development with the PROMENADE system, although it was not possible to completely adapt standard algorithms. Some parts of the system are using non-standard data objects with so many variabilities that standard algorithms could not be used.
However, using adapted standard algorithms to relate process steps and sequences of process flows together, made it possible to use standard graph-theory approaches. These approaches still need to adopt the data format within the prototype, but the global approach was already available.
As mentioned before, the only downside of the approach used is the speed of the algorithm. Data mining a large database with short sequences of process flows that need to be similar can take some time. Although a direction for future research is proposed in section 8.4 to increase the speed of the algorithm, this is still a point that needs some attention.
Except for using only standard techniques all research goals are met. Therefore this research project can be seen as successful. Especially because a working prototype is developed which can be implemented with some minor adjustments which can be read in section 8.3. The idea behind this research project together with an approach is presented at the 5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering in Ischia, Italy [LO06] and at the 11th International Conference on the Commercialization of Micro and Nano Systems in Florida, USA. In addition, a patent is filed in for this idea and approach.

# List of Figures

# 10. Bibliography


[AWO05] A. WAGENER, T. Schmidt K. Hahn R. Brück R. ; ORTLOFF, D.: Environment for Design and Verification of MEMS Fabrication Processes. In: *Proceedings of the MST 2005*, 2005. – MST 2005

[AWO06] A. WAGENER, J. Popp K. Hahn R. B. ; ORTLOFF, D.: Process Design and Tracking Support for MEMS. In: *Proceedings of SPIE: Micromachining and Microfabrication Process Technology X, San Jose* Bd. 6109, 2006. – Photonics West 2006

[BP01] BERNSTEIN, A. ; PROVOST, F.: *An Intelligent Assistant for the Knowledge Discovery Process*. http://www.stern.nyu.edu/~abernste/publ/idea.pdf. 2001. – Wrappers for Performance Enhancement in Knowledge Discovery in Databases, IJCAI

[BVL06] B. VEENSTRA, D. O. ; LANGENHUISEN, S.: An approach to exchange and generate knowledge of MEMS Process Development. In: *Proceedings of the 11th International Conference on the Commercialization of Micro and Nano Systems, Florida*, 2006. – COMS 2006

[CD97] CHAUDHURI, S. ; DAYAL, U. *An Overview of Data Warehousing and OLAP Technology*. http://www.cs.sfu.ca/CC/459/han/papers/chaudhuri97.pdf. 1997

[CK05] CIOS, K. J. ; KURGAN, L. A. *Trends in Data Mining and Knowledge Discovery*. http://isl.cudenver.edu/lkurgan/Papers/TrendsInDataMiningAndKnowledgeDiscovery.pdf. 2005

[DOV05] D. ORTLOFF, F. C. ; VEENSTRA, B.: A Systematic Approach Towards Reproducibility and Tracking of MEMS Process Development. In: *Proceedings of the 10th International Conference on the Commercialization of Micro and Nano Systems, Baden-Baden*, 2005. – COMS 2005

[Ede96] EDELSTEIN, H. *Technology How To: Mining Data Warehouse*. http://informationweek.com/561/61oldat.htm. January 1996

[Erd97] ERDMANN, M.: *The Data Warehouse as a Means to Support Knowledge Management*. http://www.dfki.uni-kl.de/~aabecker/Freiburg/Final///Erdmann/dwh_km.doc.html. 1997. – Institut für Angewandte Informatik und Formale Beschreibungsverfahren, university of Karlsruhe

[Gru93] GRUBER, T. R.: *Towards Principles for the Design of Ontologies Used for Knowledge Sharing*. http://ra.crema.unimi.it/softeng/gruber93toward.pdf. August 1993. – Knowledge Systems Laboratory, Standford

[JKP98] J.C. KRZYSTOF, R. S. ; PEDRYCZ, W.: *Data Mining methods for knowledge discovery*. 1998. – ISBN 0–79–238252–8

[JPB04] J. POPP, D. O. ; BEUNDER, M.: A Novel Approach Towards Standardization of MEMS Process Development. In: *Proceedings of the 9th International Conference on the Commercialization of Micro and Nano Systems, Edmonton*, 2004. – COMS 2004

[Kle02] KLEINBERG, K.: *An Impossibility Theorem for Clustering*. http://www.cs.cornell.edu/home/kleinber/nips15.pdf. 2002. – Department of Computer Science, Cornell University

[LH04] LI, L. ; HORROCKS, I.: *A Software Framework for Matchmaking Based on Semantic Web Technology*. http://mesharpe.metapress.com/link.asp?id=6c3xd67ppymbk8l9. 2004. – International Journal of Electronic Commerce

[LO06] LANGENHUISEN, S. ; ORTLOFF, D.: An approach to generate knowledge to support silicon based MEMS development. In: *Proceedings of the 5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, Ischia*, 2006

[Mic02] MICROSYSTEMS, Sun. *The Java HotSpot Virtual Machine, v1.4.1, A Technical White Paper*. http://java.sun.com/products/hotspot/docs/whitepaper/Java_Hotspot_v1.4.1/Java_HSpot_WP_v1.4.1_1002_1.html. September 2002

[Mis06a] MISCELLANEOUS. *Wikipedia: Clique problem*. http://en.wikipedia.org/wiki/Clique_problem. May 2006

[Mis06b] MISCELLANEOUS. *Wikipedia: Strongly Connected Components*. http://en.wikipedia.org/wiki/Strongly_connected_component. June 2006

[Shu02] SHUBIN, Sean. *Test First Guidelines*. http://www.xprogramming.com/xpmag/testFirstGuidelines.htm. January 2002

[SM98] SIMOFF, S. J. ; MAHER, M. L.: *Ontology-based multimedia data mining for design information retrieval*. http://www.arch.usyd.edu.au/~mary/Pubs/pdf/asce98.pdf. 1998. – Key Centre of Design Computing, University of Sydney

[TR01] T.H.CORMEN, C.E. L. ; RIVEST, R.L.: *Introduction to Algorithms*. 2001. – ISBN 0–26–253196–8

[WF05] WITTEN, Ian H. ; FRANK, Eibe: *Data Mining: Practical Machine Learning Tools and Technique*. Second edition. 2005. – ISBN 0–12–088407–0

# A. A run card

**Runcard (Version 1.0) Page 1 of 6**

**Process-Sequence:** Demonstrator process CAM_2
**Description:** a-Si microswitch with SiO2 sacrifical layer on nitride wafer
**Editor:** Promenade Demo User
**Author:**
**Lastchange-Date:** 2006-03-16T11:00:27Z
**Creation-Date:** 2006-01-17T15:12:28Z

| Name | Description | Type | upside | backside | Process-Params | Result-Params |
|---|---|---|---|---|---|---|
| 4inch Si Wafer (110) | | Wafer | + | + | SubstrateMaterial = Silicon<br>SubstrateOrientation =<br>SimulationWindow_Xmin = 110um<br>SimulationWindow_Xmax = 580um<br>SimulationWindow_Ymin = 525um<br>SimulationWindow_Ymax = 595um<br>MaskCell = | |
| Si3N4 LPCVD external | | ProcessStep | + | + | | Thickness = 0.3um<br>Deposited Material = Silicon Nitride |
| Cr sputtering CAM | for the bottom electrode | ProcessStep | + | - | Pressure - Ar = 2mTorr<br>Flow - Ar = 15sccm<br>Power = 200W<br>Time = 900sec | Stress = 110MPa<br>Deposition Rate = 0.24nm/sec<br>Deposited Material = Chromium<br>Conformity - Deposition = 0.<br>Thickness = 0.25um |
| **SubSequence: Lithography_AZ5214 std** | | | | | | |
| AZ5214_1.4um spin coating CAM | | ProcessStep | - | - | Rotation speed = 5000RPM | Thickness = 1.4um |
| AZ5214 pre exposure baking CAM | | ProcessStep | - | - | Temperature = 100oC @ 60sec | |
| AZ5214 standard exposure | | ProcessStep | + | - | Time = 10sec | |
| AZ5214 std development CAM | | ProcessStep | - | - | Time = 60sec<br>Concentration - AZ developer = 25Wt % | |
| D.I. Water Rinse | Wash thorougly with DI water. | ProcessStep | - | - | Time = 60sec | |
| Nitrogen Drying | Use a compresed nitrogen gas line to dry wafer | ProcessStep | - | - | Time = 60sec | |
| AZ5214 post exposure baking CAM | | ProcessStep | - | - | Temperature = 120oC @ 600sec | |
| Cr etching CAM | use chromium etchant | ProcessStep | + | + | Power = 150W<br>Flow = 50sccm | Etch Rate - Isotropic etch = 0.16666666nm/sec<br>Etched Material = Chromium |

**Runcard (Version 1.0) Page 2 of 6**

**Process-Sequence:** Demonstrator process CAM_2 **Author:**
**Description:** a-Si microswitch with SiO2 sacrifical layer on nitride wafer **Lastchange-Date:** 2006-03-16T11:00:27Z
**Editor:** Promenade Demo User **Creation-Date:** 2006-01-17T15:12:28Z

| Name | Description | Type | upside | backside | Process-Params | Result-Params |
|---|---|---|---|---|---|---|
| | | | | | Etch Rate = 1200nm/sec<br>Time - Etch = 1550sec | Etch Rate - Anisotropic etch = 0.166666666nm/sec<br>Thickness = 260nm |
| Resist strip _ Aceton | No ultrasonic both! | ProcessStep | - | - | Time = 60sec | |
| **SubSequence: Standard Cleaning** | | | | | | |
| Acetone Cleaning | Clean wafer in Acetone at room temperature, in ultrasonic bath. | ProcessStep | - | - | Time = 60sec | |
| IPA Cleaning | Clean wafer in IPA at room temperature, in ultrasonic bath. | ProcessStep | - | - | Time = 60sec | |
| D.I. Water Rinse | Wash thorougly with DI water. | ProcessStep | - | - | Time = 60sec | |
| Nitrogen Drying | Use a compresed nitrogen gas line to dry wafer | ProcessStep | - | - | Time = 60sec | |
| **SubSequence: Lithography_AZ5214 std reverse** | | | | | | |
| AZ5214_1.4um spin coating CAM | | ProcessStep | - | - | Rotation speed = 5000RPM | Thickness = 1.4um |
| AZ5214 pre exposure baking CAM | | ProcessStep | - | - | Temperature = 100oC @ 60sec | |
| AZ5214 standard exposure reverse | | ProcessStep | + | - | Time = 10sec | |
| AZ5214 std development CAM | | ProcessStep | - | - | Time = 60sec<br>Concentration - AZ developer = 25Wt % | |
| D.I. Water Rinse | Wash thorougly with DI water. | ProcessStep | - | - | Time = 60sec | |
| Nitrogen Drying | Use a compresed nitrogen gas line to dry wafer | ProcessStep | - | - | Time = 60sec | |
| AZ5214 post exposure baking CAM | | ProcessStep | - | - | Temperature = 120oC @ 600sec | |

**Runcard (Version 1.0) Page 3 of 6**

**Process-Sequence:** Demonstrator process CAM_2 **Author:**
**Description:** a-Si microswitch with SiO2 sacrifical layer on nitride wafer **Lastchange-Date:** 2006-03-16T11:00:27Z
**Editor:** Promenade Demo User **Creation-Date:** 2006-01-17T15:12:28Z

| Name | Description | Type | upside | backside | Process-Params | Result-Params |
|---|---|---|---|---|---|---|
| HfO2 sputtering CAM | insulating layer | ProcessStep | + | - | Power - RF forward = 250W<br>Flow - Ar = 25sccm<br>Flow - O2 = 2sccm<br>Pressure = 5mTorr<br>Time - Deposition = 300sec | Deposition Rate = 0.028nm/sec<br>Deposited Material = Hafnium Oxide<br>Conformity - Deposition = 0.<br>Thickness = 0.05um |
| Resist strip _ Microstripper | No ultrasonic both! | ProcessStep | - | - | Time = 60sec<br>Temperature = 80oC | |
| **SubSequence: Standard Cleaning** | | | | | | |
| Acetone Cleaning | Clean wafer in Acetone at room temperature, in ultrasonic bath. | ProcessStep | - | - | Time = 60sec | |
| IPA Cleaning | Clean wafer in IPA at room temperature, in ultrasonic bath. | ProcessStep | - | - | Time = 60sec | |
| D.I. Water Rinse | Wash thorougly with DI water. | ProcessStep | - | - | Time = 60sec | |
| Nitrogen Drying | Use a compresed nitrogen gas line to dry wafer | ProcessStep | - | - | Time = 60sec | |
| SiO2 PECVD optimized CAM | | ProcessStep | + | + | Flow - N2O = 160sccm<br>Flow - SiH4 = 2sccm<br>Flow - He = 500sccm<br>Power - RF forward = 20W<br>Pressure - Deposition Step = 1000mTorr<br>Temperature = 300oC<br>Frequency - LF-GeneratorRF = 13560000Hz<br>Purity - SiH4 = 99.999at%<br>Purity - N2O = 99.999at%<br>Purity - He = 99.995at%<br>Time - Deposition = 7500sec | Deposition Rate = 0.2nm/sec<br>Thickness = 1.5um<br>Deposited Material = PECVD Silicon Dioxide |
| **SubSequence: Lithography_AZ5214 std reverse** | | | | | | |
| AZ5214_1.4um spin coating CAM | | ProcessStep | - | - | Rotation speed = 5000RPM | Thickness = 1.4um |
| AZ5214 pre | | ProcessStep | - | - | Temperature = 100oC @ 60sec | |

**Runcard (Version 1.0) Page 4 of 6**

**Process-Sequence:** Demonstrator process CAM_2 **Author:**
**Description:** a-Si microswitch with SiO2 sacrifical layer on nitride wafer **Lastchange-Date:** 2006-03-16T11:00:27Z
**Editor:** Promenade Demo User **Creation-Date:** 2006-01-17T15:12:28Z

| Name | Description | Type | upside | backside | Process-Params | Result-Params |
|---|---|---|---|---|---|---|
| exposure baking CAM | | | | | | |
| AZ5214 standard exposure reverse | | ProcessStep | + | - | Time = 10sec | |
| AZ5214 std development CAM | | ProcessStep | - | - | Time = 60sec<br>Concentration - AZ developer = 25Wt % | |
| D.I. Water Rinse | Wash thorougly with DI water. | ProcessStep | - | - | Time = 60sec | |
| Nitrogen Drying | Use a compresed nitrogen gas line to dry wafer | ProcessStep | - | - | Time = 60sec | |
| AZ5214 post exposure baking CAM | | ProcessStep | - | - | Temperature = 120oC @ 600sec | |
| BHF Etch CAM | NH4F: HF (5:1) | ProcessStep | + | + | Temperature = 25oC<br>Time - Etch = 900sec | Etch Rate - Isotropic etch = 1.6667nm/sec<br>Etched Material = |
| Resist strip _ Aceton | No ultrasonic both! | ProcessStep | - | - | Time = 60sec | |
| IPA Cleaning | Clean wafer in IPA at room temperature, in ultrasonic bath. | ProcessStep | - | - | Time = 60sec | |
| D.I. Water Rinse | Wash thorougly with DI water. | ProcessStep | - | - | Time = 60sec | |
| Nitrogen Drying | Use a compresed nitrogen gas line to dry wafer | ProcessStep | - | - | Time = 60sec | |
| a-Si (P doped) PECVD optimized CAM | Doped with phosphine | ProcessStep | + | + | Temperature = 300oC<br>Flow - SiH4 = 50sccm<br>Pressure - Deposition Step = 300mTorr<br>Power - RF forward = 20W<br>Time - Deposition = 2800sec<br>Purity - SiH4 = 99.9999at%<br>Flow - PH3/H2 = 50sccm<br>Ratio - PH3/H2 = | Thickness = 1.5um<br>Deposition Rate = 320A/min<br>Specific Resisance = 25000Ohm.cm<br>Deposited Material = Amorphous Silicon |

**SubSequence: Lithography_AZ5214 std**

**Runcard (Version 1.0) Page 5 of 6**

**Process-Sequence:** Demonstrator process CAM_2 **Author:**
**Description:** a-Si microswitch with SiO2 sacrifical layer on nitride wafer **Lastchange-Date:** 2006-03-16T11:00:27Z
**Editor:** Promenade Demo User **Creation-Date:** 2006-01-17T15:12:28Z

| Name | Description | Type | upside | backside | Process-Params | Result-Params |
|---|---|---|---|---|---|---|
| AZ5214_1.4um spin coating CAM | | ProcessStep | - | - | Rotation speed = 5000RPM | Thickness = 1.4um |
| AZ5214 pre exposure baking CAM | | ProcessStep | - | - | Temperature = 100oC @ 60sec | |
| AZ5214 standard exposure | | ProcessStep | + | - | Time = 10sec | |
| AZ5214 std development CAM | | ProcessStep | - | - | Time = 60sec<br>Concentration - AZ developer = 25Wt % | |
| D.I. Water Rinse | Wash thorougly with DI water. | ProcessStep | - | - | Time = 60sec | |
| Nitrogen Drying | Use a compresed nitrogen gas line to dry wafer | ProcessStep | - | - | Time = 60sec | |
| AZ5214 post exposure baking CAM | | ProcessStep | - | - | Temperature = 120oC @ 600sec | |
| a-Si layer SF6 etch CAM | | ProcessStep | + | - | Pressure = 150mTorr<br>Flow - SF6 = 70sccm<br>Power = 100W<br>Time - Etch = 80sec | Etch Rate - Anisotropic etch = 20nm/sec<br>Etch Rate - Isotropic etch = 0nm/sec<br>Etched Material = Amorphous Silicon<br>Thickness = 1.6um |
| Resist strip _ Aceton | No ultrasonic both! | ProcessStep | - | - | Time = 60sec | |
| IPA Cleaning | Clean wafer in IPA at room temperature, in ultrasonic bath. | ProcessStep | - | - | Time = 60sec | |
| D.I. Water Rinse | Wash thorougly with DI water. | ProcessStep | - | - | Time = 60sec | |
| BHF Etch CAM | NH4F: HF (5:1) | ProcessStep | + | + | Temperature = 25oC<br>Time - Etch = 6100sec | Etch Rate - Isotropic etch = 1.6667nm/sec<br>Etched Material = |
| D.I. Water Rinse | Wash thorougly with DI water. | ProcessStep | - | - | Time = 60sec | |
| IPA Cleaning | Clean wafer in IPA at room temperature, in ultrasonic bath. | ProcessStep | - | - | Time = 60sec | |

**Runcard (Version 1.0) Page 6 of 6**

**Process-Sequence:** Demonstrator process CAM_2
**Description:** a-Si microswitch with SiO2 sacrifical layer on nitride wafer
**Editor:** Promenade Demo User
**Author:**
**Lastchange-Date:** 2006-03-16T11:00:27Z
**Creation-Date:** 2006-01-17T15:12:28Z

| Name | Description | Type | upside | backside | Process-Params | Result-Params |
|---|---|---|---|---|---|---|
| Butan-4-ol substitution CAM | | ProcessStep | + | + | | |
| Freeze Release CAM | Place wafer on cold plate. Fluid should almost immediately freeze. Once fluid is frozen leave 100 seconds to totally cool. Gently bring the chamber down to a vacuum, but not too quickly, else the fluid will melt. The solid will typically take 5 minutes to sublime away. | ProcessStep | + | + | Temperature = 8oC | Etched Material = PECVD Silicon Dioxide |