Abstract—As organizations consider themselves to be exposed to intense global competition, knowledge has become more and more important. A successful Knowledge Management strategy is essential for organizations in the modern business world. For this task knowledge oftentimes should be gained out of data amounts, which are not human manageable. Therefore in conjunction with Knowledge Management Data Mining is used by many organizations to transform raw data into knowledge. Among the many fields of applications the semiconductor industry is an example. In order to ensure an effective and significant analysis of the production process, a huge amount of data has to be gathered and interpreted. In this paper we present a framework processing these data. We will demonstrate how Data Mining methods could be used to support the process of finding solutions to technical problems by applying Knowledge Management.

I. INTRODUCTION

In recent years Knowledge Management (KM) has emerged as an integrated approach to achieve organizational goals by focusing on creating and managing knowledge. As companies nowadays are confronted with dynamic and changing markets as well as with continuous technological advances, organizations have realized the need for strengthening their ability to learn from current or past situations. As a result, knowledge has become an essential organizational driver and a key factor in value creation [1].

KM deals with all aspects of knowledge within the context of an organization, i.e. with its creation or capturing, sharing, storing and refining. Successful KM implementations are widespread. An overview of some examples can be found in [2]. In this paper we want to concentrate on an implementation of a Data Mining (DM) approach for KM, which is in use by a semiconductor company. However, the proposed framework is independent of application domain and would be also suitable for other areas.

This paper is organized as follows: section 2 gives an introduction to the semiconductor industry; section 3 gives a short overview about the acquisition of input data. In section 4 we present our framework in detail. Section 5 demonstrates the interaction between the framework’s components through a typical program flow and section 6 concludes this paper.

II. KM IN THE SEMICONDUCTOR INDUSTRY

Semiconductor companies develop and produce microchips, which are frequently used in everyday life. They provide products for automotive electronics as well as for industrial and consumer goods applications. As for all customers a perfect interaction with their appliances and systems is striven for, semiconductor companies have to make sure that their respective microchips are functioning in the best possible way. However, to achieve a certain level of customer satisfaction, it is not sufficient only to test the functionality of the microchips. Additionally it is inevitable to deliver the ordered microchips within the strict delivery schedules. The last demand is strongly connected with the wafer yield, which is defined as the ratio between the number of good microchips on a wafer having completed the wafer production process and the total number of microchips on the same wafer. Reaching and maintaining the planned yield level is the ultimately goal of a semiconductor company, because it is well known that the yield is strongly connected with the profit, with the ability to meet the time delivery schedules and with the quality of the microchips [3].

The just mentioned demands result in a steady effort to minimize errors in the production process. At best, sources of error should be detected at an early stage. Thus defects can be avoided before they have a chance to even develop. To reach this goal, a lot of measurements are done in the production process. As a quality gate, every parameter of every single microchip has to pass a functional test before the microchip is delivered to the customer.

This results in the recording of a tremendous amount of data during the whole fabrication process. Thereby the data are gathered on wafer level or, for an even more detailed view, on die level collected on certain positions per wafer. Anyway, the quantity of the data leads to major difficulties to analyze the data. In case of a specific production problem, it is hardly possible to identify the reasons for an unsatisfying outcome.
At this point KM comes to the fore. As stated in [4], there are three major challenges semiconductor companies are faced with: how to improve the sharing of knowledge and best practices across the organization, how to accelerate innovation rates by bringing diverse views and experience to bear, and how to quickly develop solutions to technical problems and hence reduce time-to-market. In this paper we concentrate on the last named challenge.

In the modern business world companies often have to cope with huge amounts of data, which are mostly not human manageable. Knowledge Discovery in Databases (KDD) is a research field dealing with exactly this problem and is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [5]. DM as an important step in this process is often used by organizations in order to transform raw data into usable knowledge as part of their KM initiatives. General proposals of how to connect KM and DM are presented in [6] and [7], while two concrete case studies are described in [8].

In the following sections we will show how some DM concepts, namely Neural Networks and Feature Selection, can be adopted to handle and analyze the data recorded during the wafer production process and to support an organization’s KM initiatives.

III. ACQUISITION OF DATA

After a wafer lot has gone into production, it takes about eight weeks until this wafer lot is finalized. The production is made up of several hundred processing steps. Each of these steps is generating its own data.

The monitoring of the wafer production process is based on different measurements during and at the end of the manufacturing process. Among the different kinds of measurements the ones we are interested in are the parameter test measurements (Process Control Monitoring, PCM) and the wafer function test measurements (FT). In the PCM measurements the verification of the wafer production is accomplished, i.e. the question whether everything in the wafer production has gone according to plan should be answered. For this purpose there are test structures on the wafers, which are measured electrically. These test structures are located between the microchips and called scribe test structures. They are typically depending on the production process and do not differ between products being made in the same process. So the parameters are suitable for a comparison of successful production even across product boundaries. Due to reasons concerning the throughput the parameter test measurements take place on just a sample of positions per wafer, but are accomplished on every wafer of a wafer batch. In the FT measurements the microchips on the wafers are scrutinized with respect to their functionality. One of the potential results of the FT measurements is the yield.

To ensure a useful comparison the data have to be normalized to the interval [0;1] and transformed into a predefined format as shown in the following table.

<table>
<thead>
<tr>
<th>Batch</th>
<th>Wafer</th>
<th>FT</th>
<th>PCM 1</th>
<th>…</th>
<th>PCM N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.92</td>
<td>0.65</td>
<td>…</td>
<td>0.88</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.87</td>
<td>0.62</td>
<td>…</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>1</td>
<td>24</td>
<td>0.83</td>
<td>0.71</td>
<td>…</td>
<td>0.94</td>
</tr>
<tr>
<td>1</td>
<td>25</td>
<td>0.71</td>
<td>0.77</td>
<td>…</td>
<td>0.82</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.95</td>
<td>0.81</td>
<td>…</td>
<td>0.80</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>M</td>
<td>25</td>
<td>0.98</td>
<td>0.69</td>
<td>…</td>
<td>0.76</td>
</tr>
</tbody>
</table>

This is an example for the acquisition of data on wafer level. As a wafer batch typically consists of 25 wafers, the table has 25*M rows filled with measurements. For each wafer there is a value in the FT column and a value in each of the PCM columns. The number of PCM measurements is not equal for all products; therefore we have denoted this number as N. Hence, in case of data acquisition on wafer level a single wafer batch provides 25*(N+1) measurements as input data for our system. In case of data acquisition on die level this number is multiplied by the number of dies per wafer. Our goal is to find dependencies between the values in the PCM columns and the FT column. This could be a rather simple dependency between single PCM measurements as well as complex dependencies between PCM measurements and the FT values.

IV. USAGE OF DATA MINING METHODS

DM tasks can be roughly divided into classification, clustering and association analysis. The task we are facing when trying to find dependencies between certain parameters is classification. Among the list of possible methods applicable for classification tasks, Artificial Neural Networks have emerged as one of the most frequently used classifiers. Their ability to derive useful information from complicated data has made them well suited for various prediction problems.

Feature Selection in general tackles the problem that today’s data collections have such a huge amount that even advanced computer technologies cannot prevent that the enormity of the data may cause serious problems to DM systems. Taking into account some reasonable criteria it aims at finding a minimal subset of features to allow that the original task could be achieved at least equally well compared to the full feature set [9].

In the context of a wafer production process the selection of a reduced feature subset targets the following two goals: First, it should provide the engineers an insight into a certain production problem; second, it should result in a classifier for yield prediction being as robust as possible.

As both Neural Networks and Feature Selection are predestined for our field of application, they form the two major components of our system. This is illustrated in Fig. 1.
Because of the huge amount of process steps the possible deviations in the process flow are very high. In some cases there is a dependency of many parameters, which can hardly be detected by a human specialist.

A. Feature Selection

Feature Selection is a technique to be used in the data preprocessing step of the whole KDD process. By reducing the number of features it accelerates the DM algorithm and improves the predictive accuracy as well as the result comprehensibility [10]. The number of selected features is somewhere between 0 and the total number of features, a circumstance known as the peaking phenomena [11]. Last but not least only Feature Selection makes it possible to find the root causes for an abnormal effect. Analysis of the rules, which result from a KDD run, should be human readable. This is not the fact when rules consist of about 100 terms.

The general Feature Selection process consists of four steps. The first step, the generation, has the task to build candidate subsets of features. In the second such a candidate subset is undergone an evaluation, i.e. the goodness of the candidate subset is measured and afterwards compared with the previous best. If the candidate subset is better than the previous best then a replacement of the previous best by the candidate subset takes place. To guarantee that the Feature Selection process comes to an end the third step is composed of a stopping criterion, which is interrogated at this point. The choice for a concrete stopping criterion is influenced by either the number of iterations has been accomplished. If it is based on the evaluation function, then the stopping criterion depends on whether a predefined number of features or a predefined number of iterations has been accomplished. If it is based on the evaluation function, then it depends on whether addition or removal of a feature still leads to a better subset and whether relating to the concrete evaluation function an optimal subset already has been generated. If the stopping criterion finally is fulfilled the selected subset of features is handed on to a validation.

The selection of the optimal feature subset is only guaranteed by the branch and bound method. But as this method requires a lot of computational resources, research has been focused on suboptimal methods. Among these methods the Sequential Floating Search has emerged as the method forming the best compromise between performance and computational complexity. In contrast to other methods the two values how many features are added to a certain subset or excluded from a certain subset are not fixed, but can change during to the course of the search. Additionally the feature nesting problem is solved, i.e. a feature that has been added to the feature subset can be excluded from the subset in later stages of the search.

In our system we are using the Sequential Forward Floating Search (SFFS) method presented in [12]. The exact mathematical notation of the method can be found there. SFFS is characterized by starting with an empty subset of features, i.e. it is a bottom up search. Its counterpart is the Sequential Backward Floating Search (SBFS). Fig. 2 presents the procedure as a flow chart. As shown, forward selection (FS) and backward selection (BS) are part of a great loop (indicated by the outer circle). Forward selection and backward selection are also implemented as loops (indicated by the two inner circles). In between is the checking of the stopping criterion. This is fulfilled if the current number of features i equals the number of features desired by the user.

At the beginning of the algorithm forward selection is repeated twice in order to guarantee that after the first removal of a feature at least one feature is remaining in the current subset. This is the only restriction made by the algorithm; otherwise backtracking takes place between each step of forward selection as long as the previous best subset of n features is beaten.

In order to measure the quality of a subset we are using two different evaluation functions. The first one is the Mahalanobis Distance [13], which is able to measure the distances between data points in multidimensional vector spaces. The main motivation for us to use it in our system is that it does not examine the parameters isolated, but the interplay of multiple parameters and in succession their correlations. It offers all the characteristics of the Euclidean Distance. Additionally it is scale invariant, i.e. it does not depend on the scale of the measurements. The Mahalanobis Distance is determined by calculating the inverse covariance matrix. However, in case of almost constant parameters, i.e. columns in the data table possessing of a lot of same values (extreme low standard deviation), the calculation of the inverse covariance matrix might not be possible. Therefore we are offering a k-nearest-neighbor (knn) evaluation function as an alternative solution. It is considerably slower than the Mahalanobis Distance, but is always applicable.
It takes a certain number of rows in the data table as test vectors and determines the k nearest neighbors of each test vector based on the Euclidean Distance. Each vector in the data table has a certain class membership based on the value in the FT column. The knn evaluation function compares whether the class membership of each test vector is equal to the class membership occurring most frequently among the k nearest neighbors and selects those features, which are able to maximize the output of this comparison.

B. Neural Network

Neural Networks are well known and recognized in terms of classification and approximation applicable to problems from the very simplest to complex, nearly unmanageable ones. As stated in [14] Neural Networks perform as trainable model-free estimators for high-dimensional nonlinear input-output mappings such as the ones described in this paper.

A stable, adaptive and fast learning Neural Network type is Fuzzy ARTMAP (FAM), first described by Steve Grossberg and Gail Carpenter [15]. ART refers to the adaptive resonance theory by Grossberg and Carpenter [16]; the extension MAP refers to the net’s mechanism to learn a mapping from input to output. A special feature of FAM is its growing of new neurons and weight vectors over time with respect to the complexity of the problem. This makes the net suitable for our environment, which is characterized by various and changing input data sets.

FAM fuses fuzzy logic [17] and adaptive resonance theory, taking the fuzzy subsets from fuzzy logic and category choice, resonance and learning from ART. In our system we rely on a simplification of Fuzzy ARTMAP, Simplified Fuzzy ARTMAP (SFAM) [18]. SFAM reduces the overhead of FAM without reducing its pattern recognition capability. As the net generates neurons in dependency of the input, it decides how many neurons represent a category. The absolute number of neurons depends on the input vectors, on the net parameters as well as on the number of classes.

If fed with data the input first passes the complement coder, which normalizes and stretches the input to assist the network in forming its decision regions. The resulting complement vector represents both the presence and the absence of a feature in the input.

The complement-coded representation presented to the input layer of the two layer network activates all output nodes to a certain degree. The output node with the maximum activation degree is denoted as the winner. If more than one node is chosen as maximal the node with the smallest index is announced as winner.

By the match function the complement-coded features are compared to the individual output nodes to find an output node capable of learning the presented pattern. To determine the best match to the pattern the match function is calculated for each node and compared to the vigilance parameter. If the match function exceeds this parameter the net is in a state of resonance, which proves that the node has the qualification to encode the input.

If the output node’s category matches the input category the node is able to update its weights to learn the input pattern. In the case of no match a “category mismatch” occurs and the output node’s weights remain unchanged, while the net enters the match tracking.

When match tracking starts the node with a category mismatch is shut off and the vigilance parameter is slightly increased to sharpen the criteria for the correct matching output node.

The best output node will be selected to learn the input by updating its weight vector. If no best output node has been discovered and there is no node being worthy of updating any longer, a new node is created from the input.

Fig. 3 visualizes the separate steps of the FUZZY ARTMAP training cycle to learn input patterns. For the aim of evaluating the resulting net we review the net as a set of human readable rules, in addition to the well known validation through class prediction of sample data.
This goal is achieved by transforming the nodes and weights to a very simple classification rule presentation. We benefit from the net’s transparency in decision making resulting in a very straight classification rule extraction.

V. TYPICAL PROGRAM FLOW

In the previous section we have described the working methods of both the Feature Selection and the Neural Network component. Fig. 4 demonstrates a typical program flow, i.e. a sequence of how the two components can be adopted in order to solve a specific production problem.

First of all the user has to provide some input data, i.e. he has to extract the available data of the product under analysis from the database. Beside the batches with abnormal results (unsatisfying yield) the normal ones should be extracted as well. Afterwards these data have to be split into a training and validation set. The decision how many data should be used for training is left to the user. Both a sequential splitting (first m data rows are used for training, the rest for validation) and a random splitting (m randomly selected data rows are used for training, the rest for validation) are possible and should be selected dependent on the respective analysis.

Now training and validation of the Neural Network take place. After the validation a certain prediction success rate is available, i.e. a proportional statement how many data rows in the validation set have been assigned to the correct class. A maximization of this number is striven for; if this is not possible, then the user can act on the assumption that the cause for the unsatisfying yield was not included in the input data. Otherwise he can be sure that the current dataset under observation includes the data responsible for the current investigation. However, because of the unchanged amount of the data, a conclusion to the source of error is still impossible. Therefore he can now apply the Feature Selection component. As the reduced data set should include as few features as possible, it is advisable to start with the desire for a subset consisting just of two features. Next is the splitting of the reduced data set into a training and a validation set and then the training and the validation of the Neural Network. If the prediction success rate is ok, then the user can pass over to the interpretation of the output rules. If the prediction success rate is not ok, then Feature Selection is repeated with an increasing number of desired features until the prediction success rate has reached the level necessary for the current investigation.

VI. CONCLUSION

In the first part of this paper we accentuated the need for KM in the modern business. Then we have referred to the present situation in the semiconductor industry. We pointed out that the huge amount of data recorded daily in the wafer production process results in the impracticality to analyze the data manually. Thus gaining information for a valuable decision making is almost impossible.

DM has emerged as a very promising approach in KM initiatives. By applying Feature Selection algorithms and their different evaluation functions we are able to handle the enormity of data by reducing it to a manageable amount. One of the main characteristics of Neural Networks is their applicability to multivariate non-linear problems without any further knowledge of the environment. Therefore they allow the validation of feature subsets of arbitrarily size until a minimal subset is found, which still includes the information from the whole data set.

The combination of Feature Selection and Neural Network supports KM by enabling quick solutions to technical problems. It maps the originally not manageable problem to a manageable one, and therewith connected improves and speeds up the quality control.

REFERENCES


