

Data Mining on Crash Simulation Data

Annette Kuhlmann¹, Ralf-Michael Vetter¹, Christoph Lübbling², and
Clemens-August Thole¹

¹ Fraunhofer Institute for Algorithms and Scientific Computing (SCAI),
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{kuhlmann, vetter, thole}@scai.fraunhofer.de

² BMW AG, EK-210, Knorrstrasse 147, 80788 München, Germany
christoph.luebbing@bmw.de

Abstract. The work presented in this paper is part of the cooperative research project AUTO-OPT carried out by twelve partners from the automotive industries. One major work package concerns the application of data mining methods in the area of automotive design. Suitable methods for data preparation and data analysis are developed. The objective of the work is the re-use of data stored in the crash-simulation department at BMW in order to gain deeper insight into the interrelations between the geometric variations of the car during its design and its performance in crash testing. In this paper a method for data analysis of finite element models and results from crash simulation is proposed and application to recent data from the industrial partner BMW is demonstrated. All necessary steps from data pre-processing to re-integration into the working environment of the engineer are covered.

1 Introduction

The objective of the data mining work presented in this paper is the re-use of data stored in the crash-simulation department at BMW in order to gain deeper insight into the interrelations. Here the objective is to find hidden knowledge in stored data. In principle one could think of various possible questions for such a knowledge mining analysis:

- which innovations have evolved during the design process
- were certain steps in the development unnecessary or could they be shortened
- is it possible to extract analogies between different car projects
- can reasons that have lead to certain design decisions be reproduced
- can this reasoning be applied to future projects

The data mining project in AUTO-OPT aims at examining the applicability of data mining methods on crash simulation data [1]. Due to the fact that design and development knowledge is the major asset of engineering, an automotive company cannot be expected to share large amounts of their data for research reasons. On the other hand, interesting results from data mining can only be achieved from interesting data. Therefore in this work the applicability of the

method is demonstrated, its value cannot be evaluated on the data basis available. This will be aimed at in future work.

The crash department at BMW stores all relevant information in the simulation data management system CAE-Bench [2]. Data mining queries are to be submitted from this environment. Results have to be brought back into this system and assigned to the underlying models, i.e. stored within their audit trail. This procedure is schematically shown in Figure 1.

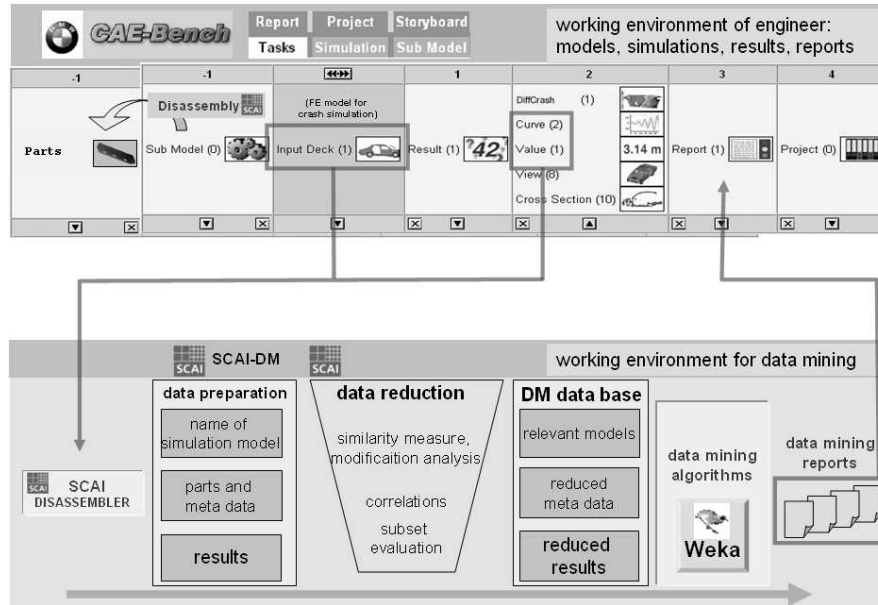


Fig. 1. Procedure for data mining. Models and crash results are exported from CAE-Bench, models are disassembled into parts, geometry based meta data are calculated, parts and meta data are stored within SCAI-DM, crash results are attached to meta data, data mining tables are assembled, DM analysis is performed, result files are produced and exported.

Fraunhofer SCAI has been provided with data from one of the most recent car projects at BMW. The vehicle under development is shown in Figure 2 (left). Each data set describes one stage of construction of this car within the development process via finite element models (FE models) made up by about 500.000 independent nodes and elements.

Each car is composed of app. 1200 parts. CAE-Bench stores the models as complete vehicles, i.e. one single large FE model, called input deck. In order to analyse the geometry of the parts these input decks need to be disassembled as shown in Fig. 2 (right) for an older BMW vehicle, the new model cannot be shown in such detail because of a nondisclosure agreement.

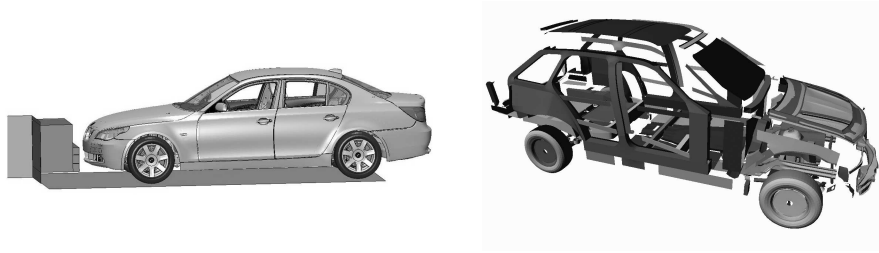


Fig. 2. Recent model of BMW employed for data mining (left). One FE input deck consisting of numerous parts (right).

2 Preparation of the data for data mining

It is generally accepted that the preparation of the data involves as much as 80-90% of the effort when a data mining task is attempted, see e.g. [3]. The data cannot be processed by a data mining tool in their original format. To the authors best knowledge no approach for data mining on raw finite element data exists. The preparation of the data thus constitutes the main challenge for the data mining approach on the FE data. In addition, the data has to be cleaned and checked for consistency and the appropriate values have to be combined. As a first but major step a process for data preparation has been developed:

- a) Export of data from CAE-Bench
- b) Disassembling into parts and computation of meta data
- c) Data cleaning and sorting — clustering of parts
- d) Similarity analysis and data reduction — clustering of variants
- e) Evaluation and cleaning of crash result data

As a result of this procedure a table is generated that allows for access to the data with data mining algorithms. This section focuses on the preparation of the data, whereas the application of the data mining algorithms will be presented in section 3.

a) Export of data from CAE-Bench

CAE-Bench can export selected input decks along with the result achieved when these models were subjected to a virtual crash test. An example is shown in Fig. 4. Information is extracted from this export, such that the relevant crash results can be attached to the respective input deck data and stored in the SCAI data mining framework (SCAI-DM).

b) Disassembling into parts and computation of meta data

Motivation. The data mining approach in this work concerns the shape of the parts of the car. The aim is to analyse how changes in shape have influenced crash

behaviour. The FE-model itself contains all geometrical information. However, this information is hidden from data mining algorithms, as these cannot extract meaningful knowledge from node and element descriptions. Meta data has to be determined such that it quantifies geometry in an appropriate manner. In this work several values have been chosen as meta data, e.g. the centre of gravity of each part, the moments of inertia, the length of edges and margins, surface size, bounding box, length of branching lines—as shown in Figure 3. All of these are mesh independent. They thus enable comparison between models that have been meshed with different algorithms or programs. The meta data reduce the amount of data massively, such that handling of data is facilitated considerably.

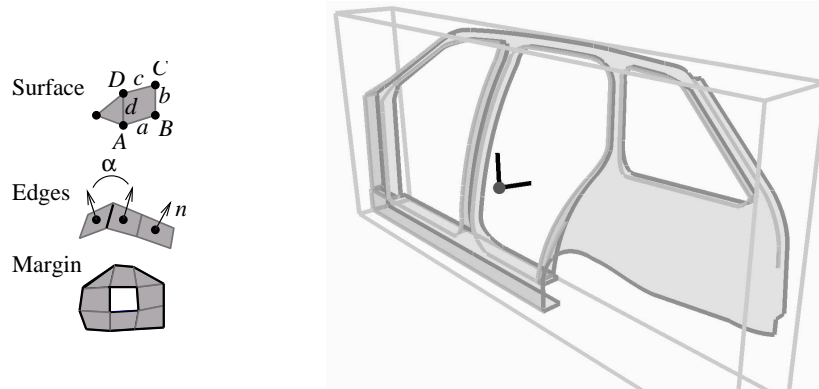


Fig. 3. Typical meta data and their appearance in an example part.

Reading the Input Deck. Today the body-shell of a finite element car model is described by an input deck of 100 MB containing approximately 1.500.000 lines. Figure 4 shows a small subset of such an input deck. The part number indicates which element of the meshes belongs to which specific part of the car. The material section defines a homogeneous density and thickness for each part. In the disassembling procedure all elements with the same part number and their respective nodes are extracted from the input deck and form one new mesh for this single part.

For each part the disassembler thus extracts a sub-mesh of the input deck. This sub-mesh is the basis of the calculation of the meta data. The sub-mesh files are also used to create previews of the parts: from three different angles or—on demand—in form of a three-dimensional applet visualisation [4–6]. Since the generation of previews of parts is a time intensive process, it is initiated only for new parts which have not previously been stored in the database.

Computation of Meta Data. For meta data calculation various details on FE models have to be taken into account. The model surfaces here are curved.

Node coordinates (x,y,z)						
NODE /	1000000	-184.0	-370.475	138.371	0	0
NODE /	1000001	-46.178	-366.451	126.769	0	0
NODE /	1000002	-46.178	-366.451	117.748	0	0
NODE /	1000003	-164.298	-370.252	136.719	0	0
NODE /	1000004	-144.609	-369.677	135.061	0	0

Node connections (quadrilaterals or triangles)								
SHELL /	1004043	11081	1040229	1025655	1025656	1040230	3	0.0
SHELL /	1004044	11081	1001494	1025655	1040229		3	0.0
SHELL /	1004045	11091	1000014	1000015	1000000	1000003	3	0.0
SHELL /	1004046	11091	1000013	1000014	1000003	1000004	3	0.0
SHELL /	1004047	11091	1000012	1000013	1000004	1000005	3	0.0

Part numbers						
MATER /	11081	103	0.39590890E-07			
ZSTE300E_MTT_vo_sussen_li						
2.070E+05CURVE		3.000E-01	1.80	1.000E-02	1.000E-02	1.000E-02 0.000E+00
MATER /	11091	103	0.39590890E-07			
ZSTE300E_MTT_vo_innen_li						
2.070E+05CURVE		3.000E-01	1.80	1.000E-02	1.000E-02	1.000E-02 0.000E+00

Material density and thickness						
			1.80			

Fig. 4. Excerpt of an input deck: the NODE and SHELL sections describe the geometry of the finite element meshes; in the MATER section the thickness d and density ρ of the material can be found. Beside shell elements, which can be triangles or quadrilaterals, additional elements like membrane elements (with 4 nodes), solid (8 nodes), beams (2 nodes), or bars (3 nodes) appear in an input deck.

Using shell elements for the description means that the four corner points of a quadrilateral do not necessarily lie in one common plain, see Fig 3. One well defined way to calculate their surface is $S \approx \frac{1}{2} [(\mathbf{a} + \mathbf{c}) \times (\mathbf{b} + \mathbf{d})]$. The mass of an element is given by its surface multiplied by the material thickness d and density ρ . The centre of gravity, at which the mass m is assumed to be located in a single element is positioned approximately at $\frac{1}{4} [\mathbf{A} + \mathbf{B} + \mathbf{C} + \mathbf{D}]$, where $\mathbf{A} \dots \mathbf{D}$ are the corner points of the SHELL element. Then the centre of gravity and the moments of inertia of the complete part are given by their sum over all point masses. For every element a normal vector is constructed by $\mathbf{n} = \frac{\mathbf{a}+\mathbf{c}}{|\mathbf{a}+\mathbf{c}|} \times \frac{\mathbf{b}+\mathbf{d}}{|\mathbf{b}+\mathbf{d}|}$. The normal vectors \mathbf{n}_1 and \mathbf{n}_2 of adjacent elements are used for detecting edges. If the angle $\alpha = 2 \arcsin(|\mathbf{n}_2 - \mathbf{n}_1|/2)$ is larger than a user defined value, the connection line between the elements is called an edge of the mesh. If one side of an element is not connected to any further element, this line is assumed to be a margin of the structure. In this manner all meta data characterising the geometry of each part is computed.

c) Data cleaning and sorting — clustering of parts

The finite element models are subject to numerous kinds of modifications. During the engineering process in which a car model is improved with respect to its crash-worthiness a subset of parts is modified. In general the parts modified are the crash-relevant ones. Additional modifications follow the demands of other engineering disciplines, e.g. holes may be inserted into the parts in order to achieve a better drain of varnish during production. Such measures reduce crash-worthiness, which then again has to be improved by further modifications.

		Model	Number	Mass	Margin	Surface	Edges	MD5Sum
side view	top view	E6016VS01_fd01v02	11171	3.72113E-4	2323.04	49221.4	473.896	ed6bee68c9d11dcfb09d12839d7
Fraunhofer SCAI								
side view	top view	e60ppp_fd08v11a	11171	5.42834E-4	2009.7	52771.9	480.21	c261dc25558729900093c57f25d
Fraunhofer SCAI								
side view	top view	E60PPP_fd01v01a	11171	5.89499E-4	1469.6	57328.8	757.541	4d5647dbd0ceedb6b213ba89df
Fraunhofer SCAI								
side view	top view	e60ppp_fd01v01b	11171	5.89499E-4	1469.6	57328.8	757.541	4d5647dbd0ceedb6b213ba89df
Fraunhofer SCAI								

Fig. 5. Screenshot of SCAI-DM framework, here some variations of part no. 11171 with meta data and checksums. The database contains about 146.000 parts belonging to 134 different crash tests. Deleting all sub-mesh files with multiple MD5 checksums (right column) the database can be reduced by 93% to 9900 different parts.

However, not all parts are modified in all stages of car design. As unchanged parts cannot be responsible for deviations in the simulation results, such parts can be excluded from the analysis. In order to remove the parts that never have been modified MD5 checksums are created for all sub-mesh files, Fig. 5 (right column). If the checksum of any part stays constant in any data set of interest the part was left unchanged and one single reference of the sub-mesh file is stored. Solely parts with more than one instance in the data base are included in the data mining queries.

SCAI-DM Data Base. After disassembling all parts are stored in the SCAI-DM framework along with their meta data, as shown in Fig. 5. Depending on the purpose parts and data can be displayed in any other combination.

Avoiding Inconsistent Naming/Numbering. One bottle neck for data mining of the BMW data is the fact that text entries in the data management system are free text. Some agreements are complied with in the majority of cases. Repeatedly, however, re-naming and re-numbering of parts was encountered in the data, which showed that rules were not consequently followed. Therefore, to avoid irrelevant results from the analysis aimed at it is vital that all data entering the analysis stick to the same rules. The safest way to achieve correct data is to avoid the text entries in CAE-Bench altogether and use the FE descriptions as

a basis. This again implies that an automatic method to identify parts has to be set up such that the use of part-numbers or -names coming from CAE-Bench is avoided. The meta data calculated from the FE model can be the basis for part identification using cluster analysis. The clustering process divides a dataset into mutually exclusive groups such that the members of each group are as "close" as possible to one another, and different groups are as "far" as possible from one another, where distance is measured with respect to all available variables, see e.g. [7].

Each meta data property spans a new dimension in the similarity space. The meta data of a specific part are represented by a point in the multidimensional similarity space. Similar parts have similar meta data and form a cloud of adjacent points. Figure 6 (left) shows the idea of clustering of meta data in a schematic diagram: Two dimensions of the similarity space are shown. The meta data of the parts L and C form two clouds, in which a substructure indicates the presence of several modifications. Figure 6 (right) shows a clustering plot of the BMW data. The dots describe to two different parts with the same part number 11011 "Motorträger" (three clusters on the right) and "Schottblech Motorträger" (small cluster in the centre of the diagram). This is an example for a change in numbering of a part.

d) Similarity analysis and data reduction — clustering of variants

Differing checksums indicate that a sub-mesh was modified in some unknown way. Then negligible file modifications have to be distinguished from relevant changes such as modified shapes. In this framework for data mining the geometric meta data, as described above, serves as a similarity measure for the parts. Minor and major changes of the parts design will result in a hierarchical structure of clouds and sub-clouds, see Fig. 6 (left). Using hierarchical clustering a substructure can be found inside the clusters. Starting with C_2 as a reference, C_1 contains parts with a higher mass (caused by a higher thickness d or density ρ of the material) while the parts in C_3 result from geometrical modifications increasing the surface (e.g. caused by additional beadings for higher stiffness). In the clustering plot of BMW data, Fig. 6 (right), the light grey dots belong to three modifications of the same part, namely 11011 "Motorträger". An example for typical modifications and their influences on the meta data can be seen in Fig. 5, where similar parts have been selected from the data base.

This clustering of parts in the meta data space in order to identify variants of designs is a time consuming task when all relevant parts and meta data are considered. An alternative method leading to similar results is to merge the meta data into a single similarity measure [8]. For the work presented in this paper a weighted sum of all the meta data has been employed. Then, if the weights are appropriately chosen, parts with the same similarity measure are similar in shape. This similarity measure serves as the main attribute for data mining, as described in section 3.

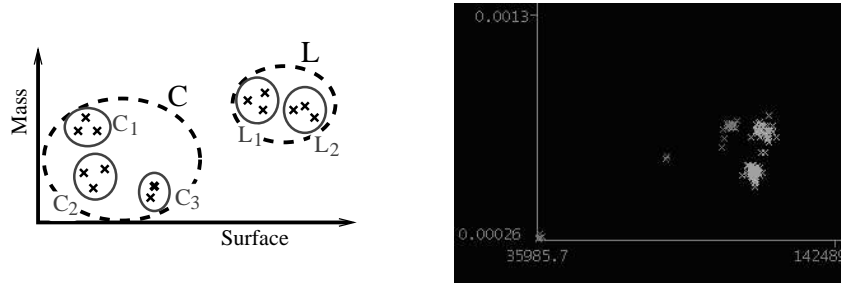


Fig. 6. Left: idea of hierarchical clustering in a schematic diagram. Two dimensions of the similarity space are shown. Right: meta data of real parts. The grey clusters on the right correspond to three different variants of the same part.

e) Evaluation and cleaning of crash result data

For each crash simulation several values and curves, as well as images and movies are computed in order to evaluate the crash worthiness of this particular design. The bottle neck here is similar as before: the scripts that calculate the values stored in CAE-Bench can be altered at any time, such that the compatibility of the values has to be ensured before data mining can be attempted. No automatic approach could be developed to check this compatibility so far. In this work values whose scripts have been left unchanged for all simulations have been used for the DM analysis. This could, however, be a serious drawback of the method and other possibilities of ensuring reproducible values for the crash results have been discussed with BMW. In this paper the only result values analysed are intrusions. Intrusions measure the difference between the distances of two points inside the car (one FE node) before and after the crash test.

In the last step of data preparation the data base is reordered. A table containing one line per crash test is formed: the name of the model, the similarity values of the parts and the result values of interest.

3 Datamining on similarity data

The aim of this work is to evaluate the applicability of data mining methods for simulation data in engineering. As a result from a complex data preparation procedure a table suitable for data mining can be achieved in which simulation data appears transformed into geometrical meta data. This table (Fig. 7) is written in Weka format, for which readily applicable data mining algorithms are available, see [11, 12].

3.1 Attribute Selection

An important step in data mining is the selection of those attributes that are relevant predictors before starting to build the model [9]. This is important because too many may be available when the full data set is encountered. Irrelevant

Model	Sim_8010	Sim_11011	Sim_11012	Sim_11013	Sim_11014	Sim_11021	Sim_11022	...	Intrusion 1	Intrusion 2
E6016V501_fd01v02	0.13	3.11	3.11	0.91	0.91	0.53	0.53	...	-252.578	-217.411
E6016V501_fd01v02a	0.13	3.11	3.11	0.91	0.91	0.53	0.53	...	-248.203	-217.2
e60vbg1+_fd02-1v06	0.0	1.13	1.0	4.08	4.02	0.01	0.01	...	-110.295	-114.728
e60vbg1+_fd02redv05	0.0	1.29	1.0	4.05	4.02	0.01	0.01	...	-191.374	-180.895
e60vbg1+_fd02v04	0.0	1.29	1.0	4.05	4.02	0.01	0.01	...	-118.985	-137.677
e60vbg1+_fd02v04b	0.0	1.29	1.0	4.05	4.02	0.01	0.01	...	-116.046	-126.97
e60vbg1+_fd02v07	0.0	1.13	1.0	4.08	4.02	0.01	0.01	...	-117.73	-123.492
e60vbg1+_fd03-1v09	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-155.667	-169.425
e60vbg1+_fd03-2v10	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-139.427	-145.34
e60vbg1+_fd03-2v10a	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-148.988	-163.872
e60vbg1+_fd03-2v10b	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-156.212	-170.325
e60vbg1+_fd03-3av12	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-148.141	-146.928
e60vbg1+_fd03-3bv12b	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-142.861	-162.149
e60vbg1+_fd03-3v11	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-146.973	-153.957
e60vbg1+_fd03-4v13	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-142.953	-145.71
e60vbg1+_fd03-5v02	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-158.559	-166.717
e60vbg1+_fd03v08	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-117.151	-126.494
e60vbg1+_fd04v14a	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-148.246	-173.806

Fig. 7. Re-ordered table for data mining: each line contains one model with similarity values of relevant parts and serves as instance for analysis. Simulation results (intrusions) are attached and serve as classes.

information should be excluded from the data set [10]. Thus a feature selection algorithm can show which attributes have the strongest influence on the class. For the crash simulation data this information can be particularly valuable, as it reduces a vast amount of geometrical modifications to a small number of seemingly important ones.

Employing an attribute selection algorithm on the crash data, e.g. ChiSquared in Weka, means that parts are ranked depending on the impact of the variation of their similarity measure on the intrusion of interest. In Figure 8 one result of such a calculation is shown. The list of parts shows those 6 out of 1200 whose variations have most influence on the intrusion. In this case the data basis is 30 models—the portability therefore is likely to be rather limited.

Attribute Evaluator (supervised, Class (nominal): 351 value_Some_Intrusion: Chi-squared Ranking Filter		
Ranked attributes:		
33.5022	1	Sim_11013
28.02819	5	Sim_15121
21.1853	148	Sim_11032
21.1853	146	Sim_11042
21.1853	147	Sim_11041
21.1853	149	Sim_11031

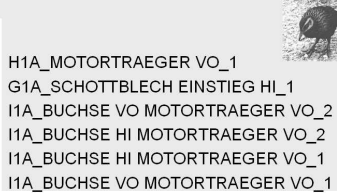


Fig. 8. Attribute selection with Weka: The six parts whose variations have most influence on the intrusion. BMW has confirmed the importance of these parts for the front crash simulated here.

3.2 Decision Trees

Another method employed in order to demonstrate the possible outcome of data mining on simulation data is the decision tree method. Here a further step has

been taken towards the achievement of results relevant to the application engineer. In practice the engineer is very rarely interested in the behaviour of only one of his result values, instead he needs to get an understanding of the influences of his design modification on a range of values. For this reason four result values were selected and clustered into three groups, one of which covers the most desired vehicle behaviour during crash. The clustering of the instances into three groups is demonstrated in Figure 9 for two of the result values. A clear grouping into "good" (circles), "medium" (squares) and "poor" (triangles) crash tests can be seen.

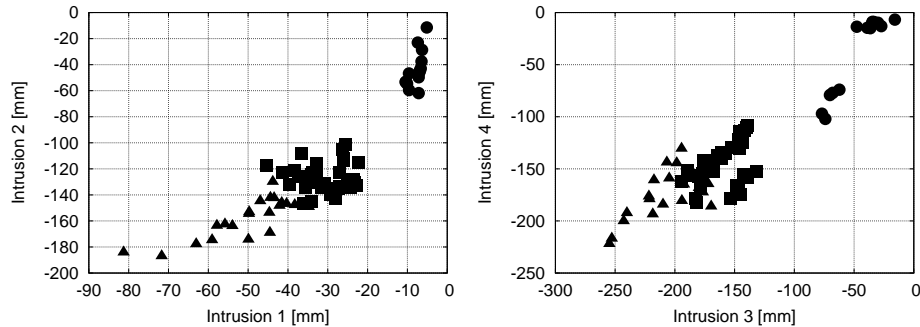


Fig. 9. Clustering of result values "Intrusion 1 . . . 4" in order to be able to represent various aspects of crash behaviour with one single class value.

The membership of a model to these clusters is then used as "class" when the decision tree is built. As attributes the similarity measure—in this case again a weighted sum of all meta data—is employed.

An example for such a tree is shown in Figure 10. The tree thus shows in which cluster a car model can be expected to lie depending on the geometrical version of the parts contained in the car model. These represent now the nodes of the tree.

For this example a data basis of 77 crashtests has been employed, which still is a rather small basis for rule building. However, these results are promising because the similarity measure seems capable of adequately representing shape modifications and lead to meaningful results.

3.3 DM Reports

The results achieved within the SCAI-DM framework is imported into CAE-Bench in order to be accessible by other engineers at other times. The reporting tool in CAE-Bench can include text and figures, such that a data mining report can be stored with the underlying input decks in CAE-Bench. This closes the circle of the procedure shown in Fig. 1.

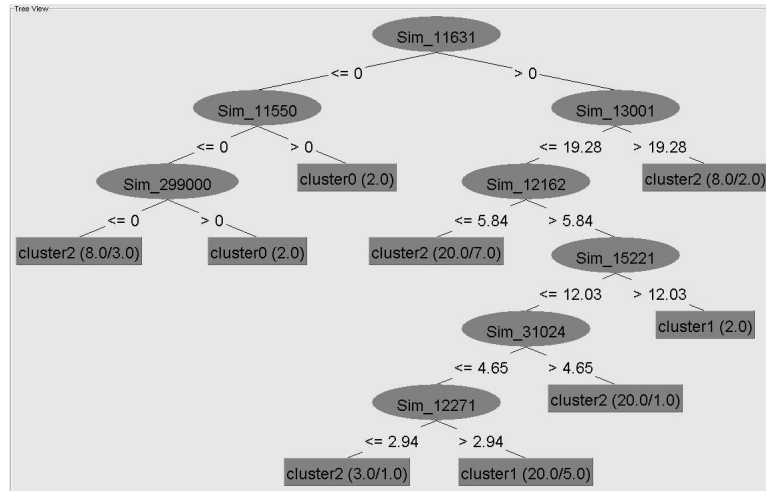


Fig. 10. Decision tree `weka.classifiers.trees.J48 -C 0.25 -M 2`. The existence of the part 11631 "Abstützung Lenksäule Unterteil" is the determining factor for the intrusions. If this part was integrated (> 0) further important parts in this data basis are two modifications of part 13001 "Bodenblech vorne" and part 12162 "Verbindung Längsträger".

4 Results

The applicability of data mining on crash simulation data has been demonstrated in this work. A framework for data preparation has been developed. The computation and handling of meta data for similarity search has been studied in detail. The employed similarity measure has proved to be appropriate for detection of relevant changes in shape. The usability of the approach on data from an automotive application has been shown. Due to the limited amount of data available for this work conclusions are limited, but first significant results have been achieved on a test set of data. The next step aimed at is the integration of selected algorithms and data preparation tools into CAE-Bench. As soon as this has been accomplished the method needs to be validated on a more substantial data set, i.e. within the working environment of BMW. Then it will be feasible to judge whether the original questions aimed at can be answered.

Acknowledgements

We wish to thank T. Gholami from BMW AG for his support of this work.

References

1. Verbundprojekt AUTO-OPT, funded by the German Ministry for Education and Research, www.auto-opt.de

2. CAE-Bench by MSC.Software, Munich, www.mscsoftware.com
3. Kloesgen, W., Zytkow, J., (Eds.): Handbook of Data Mining and Knowledge Discovery. Oxford University Press (2002)
4. Borwein, J., Morales, M.H., Polthier, K., Rodrigues, J.F., (Eds.): Multimedia Tools for Communicating Mathematics. Springer Verlag (2002) ISBN 3-540-42450-4
5. The JavaView Project, www.javaview.de
6. Keim, D.A.: Datenvisualisierung und Data Mining. Datenbank-Spektrum **2** (2002) pp. 30-39
7. Kriegel, H.-P., Brecheisen, S., Januzaj, E., Kröger, P., Pfeifle, M.: Visual Mining of Cluster Hierarchies. Proc. 3rd Int. Workshop on Visual Data Mining, Melbourne, FL, (2003) pp. 151-165.
8. Kriegel, H.-P., Kroger, P., Mashael, Z., Pfeifle, M., Potke, M., Seidl, Th.: Effective Similarity Search on Voxilised CAD Objects. Proc. 8th Conf. on Database Systems for Advanced Applications, Kyoto, Japan (2003)
9. Devaney, M., Ram, A.: Efficient Feature Selection in Conceptual Clustering. Proc. 14th Int. Conf. on Machine Learning, Morgan Kaufmann Publishers (1997) pp. 92-97
10. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence, **97** (1997) pp. 245-271
11. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000)
12. Weka 3: Data Mining Software in Java, www.cs.waikato.ac.nz/~ml/weka
13. Kuhlmann, A., Thole, C.-A., Trottenberg, U.: AUTOBENCH/AUTO-OPT: Towards an Integrated Construction Environment for Virtual Prototyping in the Automotive Industry. Lecture Notes in Computer Science **2840** (2003), pp. 686-690