

# Integration of Morphing and Optimization with the CAX-Load Case Composer at AUDI

Dr. Holger Meissner<sup>1</sup>, Marko Thiele<sup>2</sup>

<sup>1</sup> AUDI AG, Ingolstadt, Germany

<sup>2</sup> DYNAmore GmbH, Stuttgart, Germany

## Summary:

The increasing demand to evaluate vast amounts of different load cases has led to a highly standardized and automated way of model assembling at AUDI. This model assembling is greatly assisted by the software "CAX Load Case Composer" which has been developed by DYNAmore in cooperation with AUDI. The **Loadcase Composer** (LoCo) provides the user with convenient ways to manage FE-model include files and allows to automatically select appropriate include files for each load case. Thus the use of redundant includes can be avoided or at least reduced by significant amounts. One mayor concept of LoCo is the capability of integrating parameters in the FE input files. Parameters for design changes, such as for example airbag settings or seat/dummy transformations can be specified and administrated within the software. This allows the user to apply parameter studies, optimization and stochastic analysis very fast and easily. Through the integration of LS-OPT in LoCo, powerful optimization algorithms can be employed.

In this paper it is shown how morphing parameters for geometrically shape changes have been integrated in LoCo. This will be demonstrated with an example. The close integration into the standardized simulation workflow allows performing parameter studies of shape design changes with a minimum effort. In addition, it can be used in conjunction with the LS-OPT integration in LoCo. Together with LS-OPT and LoCo an engineer at AUDI has the ability to set up an optimization with very little effort. Thus it allows that optimization, parameter studies and stochastic analysis become operations of daily use.

## Keywords:

Morphing, CAE process automation, shape optimization, FE-model assembling

## 1 Introduction

No more than 5 years ago a full vehicle crash model consisted of just one large input file. Today the models are designed in a modular structure. The complete Finite-Element input file for the crash solver is assembled based on many different include files, which consist of model components, such as airbags, seats, doors, dummies, etc. This allows for using the same parts of a model in different load cases whereas only some parts have to be changed in order to adapt for differed scenarios. Figure 1 shows a simplified assembling of a full crash model.

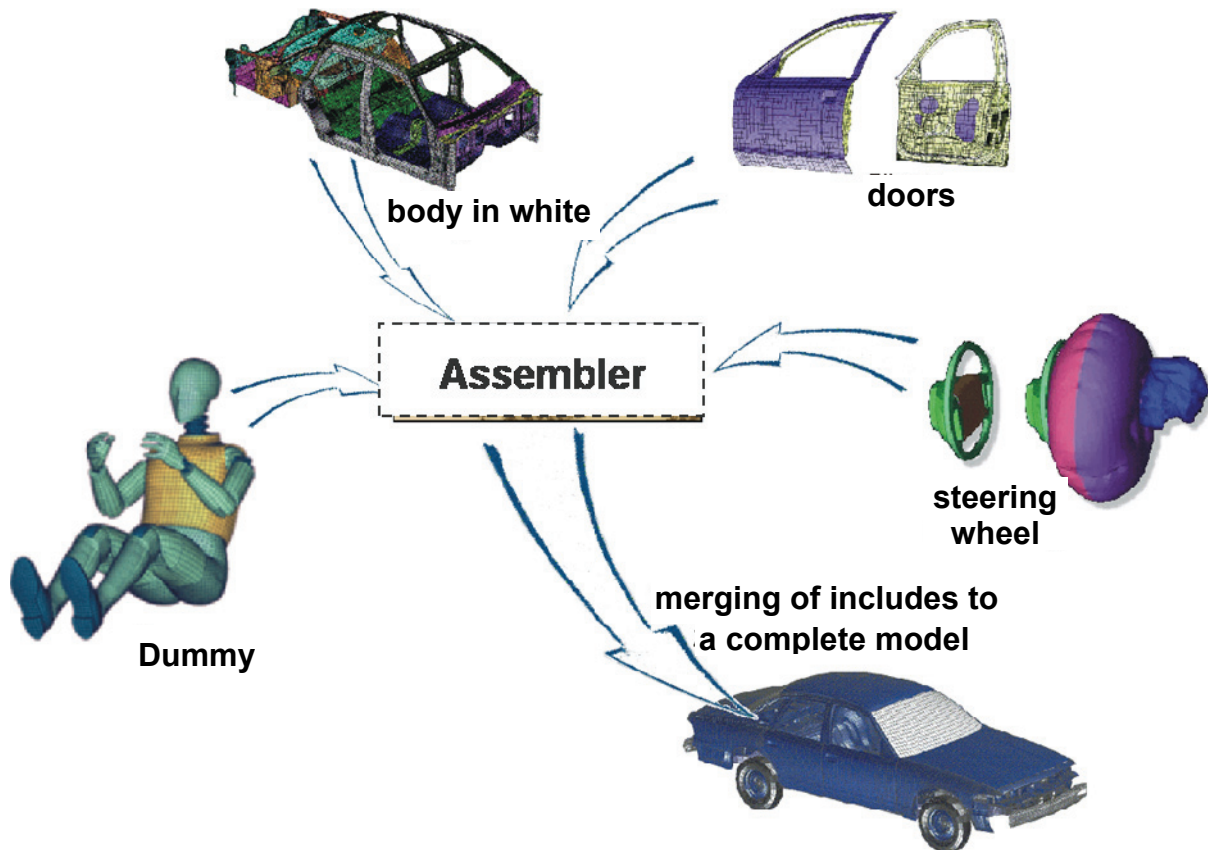


Figure 1: Simplified assembling of a full car crash model based on a set of include files

A mayor challenge assuming such an approach is the management of all these include files in a multi-user environment and the automated set-up of several load cases simultaneously. In order to accomplish this, AUDI actively developed a kind of **Simulation Data Management (SDM)** System in cooperation with DYNAmore which is called "CAX Loadcase Composer" (Figure 2). Other than traditional **Product Data Management (PDM)** systems its focus is mainly on the CAE processes.

Since FE simulations are very complex with mostly ASCII files involved numerous different preprocessors are used ranging from highly sophisticated meshing and morphing tools like ANSA all the way down to text editors like vi, nedit or emacs. The usual CAE engineer seems to spend most of his time editing includes with a couple of preferred preprocessors and possibly uses for each task a preferred tool. Therefore, the most important tool for the CAE engineer is still the shell from where he can access all the tools and user defined scripts instantly.

One main aspect of the development process of LoCo is to consider these facts and not to limit the engineers rather than support them. Therefore the tool can be used either from the shell and the file system or using a thin client which provides a sophisticated GUI and an optimized workflow (Figure 2).

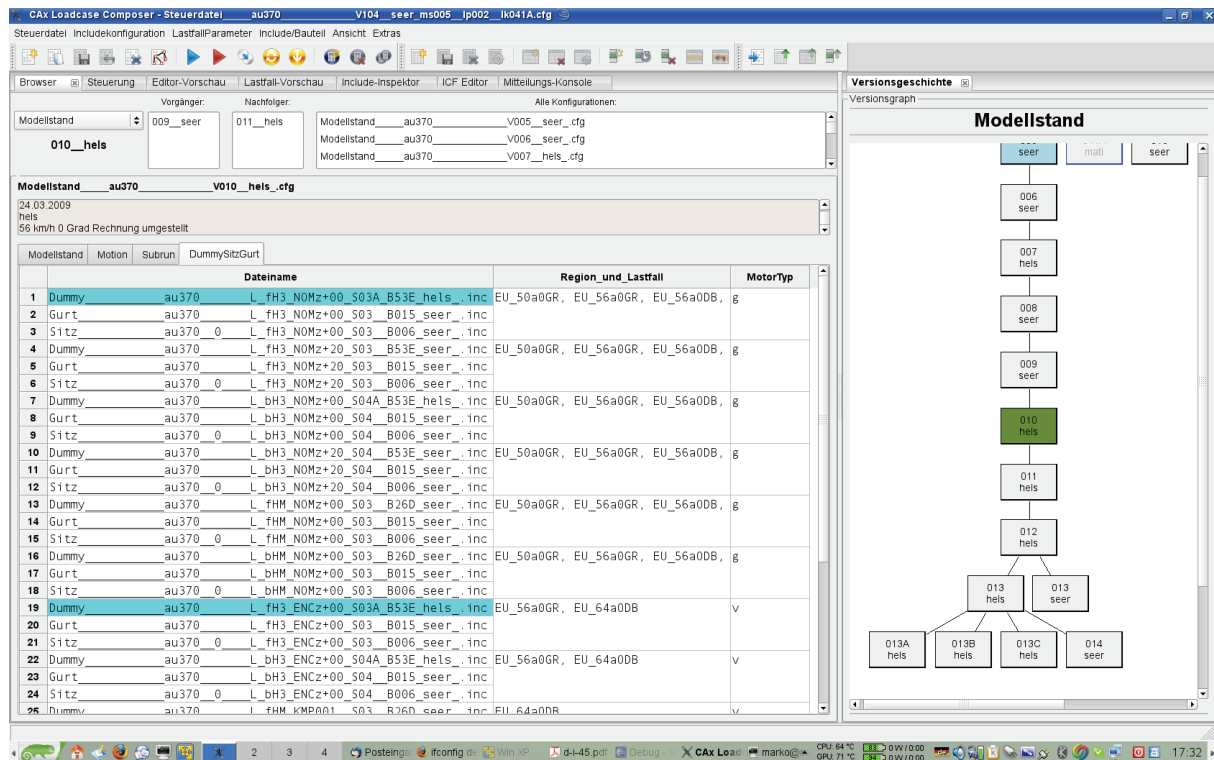


Figure 2: GUI of LoCo showing the browser and history view

## 2 Model assembly using LoCo

### 2.1 Requirements

The CAE department of a car manufacturer usually is divided into several different crash disciplines represented by different engineering teams. For example, there are teams working on body in white crash, occupant safety crash analysis for front, side and rear impact load cases, pedestrian safety and impact analysis for interior components. Regarding solutions for the management of simulation data and process work flow there are different basic requirements. The main points can be summarized as follows:

- visualization of available model components (include files) and their properties
- Automated assembly of the full FE-model based on sub-models for several load cases simultaneously.
- Verification and check of the assembled full model according to the requirements of the respective Finite-Element solver.
- Local copies in the working directory should be possible. Users want to work with the include files and the full FE-model on their local disc with their local tools. Therefore, synchronization of the local and global data is necessary. Global data are available to other users of the team, local data not.
- Supply of a Multi-User and Multi-Discipline environment. Consideration of different discipline specific requirements and work flows.
- Trace back of the history of the include files and load case variants. Detection of parent and child configurations. Comparison of variants.
- Standardized and automated documentation of modifications and of new units.
- Integration of CAT disciplines in order to synchronize activities with the testing departments.
- Monitoring of the project status.
- Open and flexible system, that allows integration of arbitrary external software. Engineers want to work with their favorite tools for pre- and post processing.
- Support of parameterization of input files in order to launch parameter studies easily (integrated optimization and DOE-studies).

In collaboration with AUDI AG a new software tool “CAX-LoadCase Composer” (LoCo) has been developed during the past years. The software takes into account the requirements, which are listed above. The realization of LoCo is based on a client-server architecture interfacing to a centralized data storage and management system (see also Figure 6). Typically users work with local working copies of the data from one project. Once a variant is finalized and optimized with respect to specific load cases the modified components are fed back to a sub-model pool available to all other users. Furthermore, the history of the modifications is recorded and stored for documentation. History trees of the variants generated during the development process of a vehicle series can be visualized and properties, such as mass, can be compared (Figure 3). There are different zoom levels for the visualization in the graphical user interface that control the level of detail and the type of information shown in the GUI.

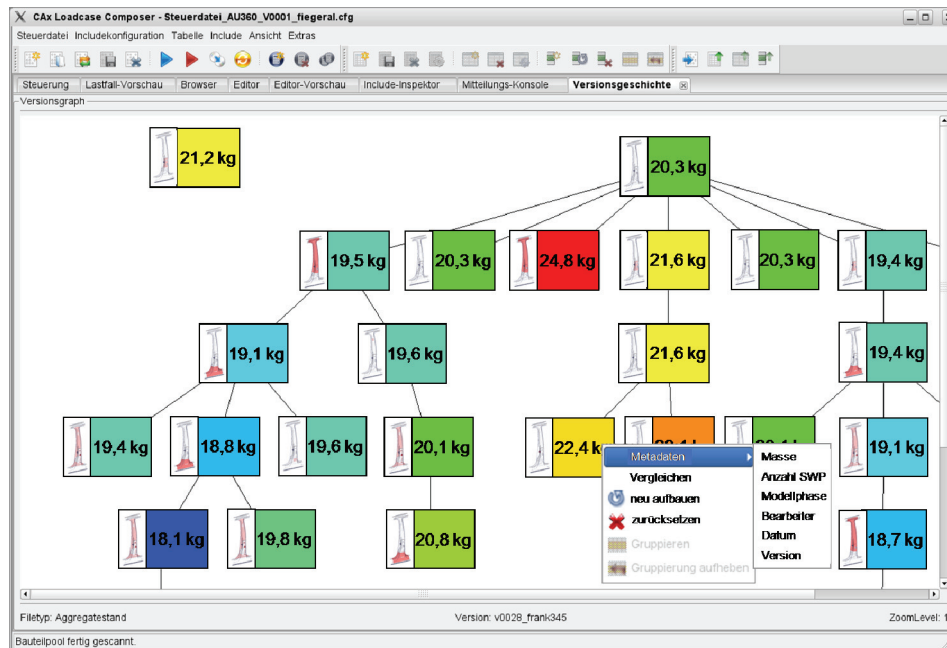


Figure 3: History of simulation model variants with mass comparison [1].

## 2.2 Parameterization of models

A major aspect in LoCo is the capability of integrating parameters in the FE input files. Often only a parameter has to be changed within an include in order to prepare it for a specific load case. Parameters for design changes, such as for example airbag settings or seat/dummy transformations can be specified and administrated within the software environment and assigned to individual load cases. This allows the user to apply DOE-Studies, optimization and robustness analysis very fast and easily. In LoCo powerful optimization algorithms, provided by the commercial optimization software LS-OPT, are integrated (see Stander et al [3]).

For visualization and post-processing of results, coming from optimization and robustness studies, the software tool D-SPEX (**D**esign **S**pace **E**Xplorer) has been developed [4]. Its primary focus is on the visualization of meta-models for design studies in order to help the engineer to understand effects of design parameters with respect to specific responses (Figure 4). Other features of D-SPEX include statistical evaluations, even of time history curves, such as mean values, standard deviations, correlation indicators and ANOVA (Analysis of Variance) results. In near future D-SPEX will be integrated completely into the LoCo environment.

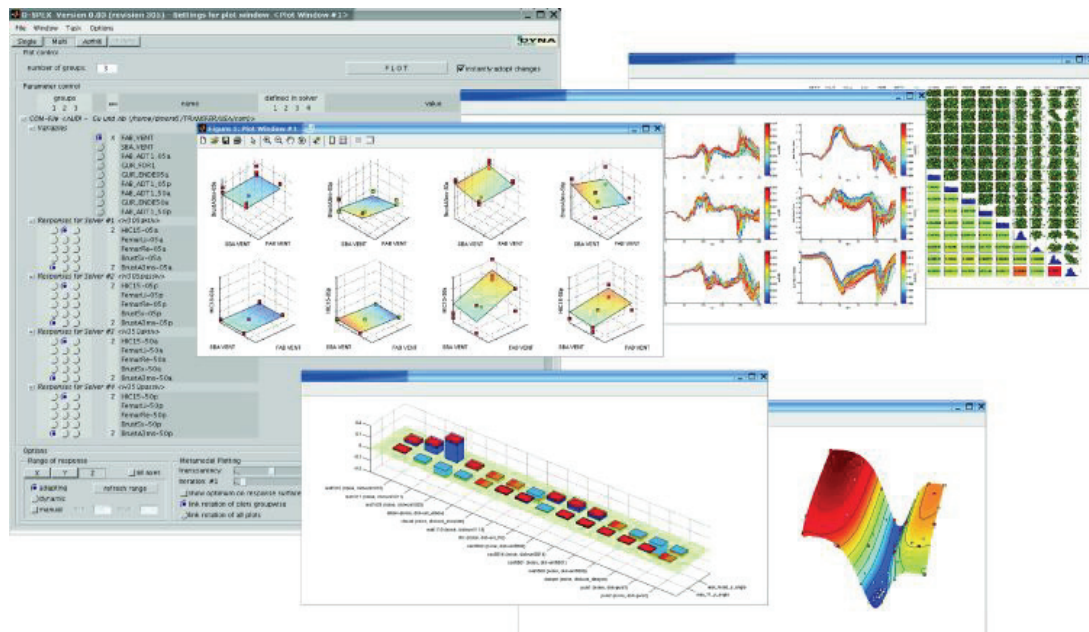


Figure 4: Integrated DOE-Studies and optimization analysis based on Meta-Model approaches such as Radial Basis Functions (RBF's) and Neural Networks.

### 2.3 Template based model assembly

Once all the necessary includes and parameters for one or more load cases are gathered by LoCo, the includes are to be assembled into a complete model. This assembly process is done in LoCo by several means. On the one hand the LoCo provides standard assembly routines that just merge the includes into one complete model and replace all the parameters in the include files. On the other hand LoCo provides an interface to user build scripts or assembly templates via a template language. The user may prepare these scripts or assembly templates using a special template language [7] in order to receive from LoCo the necessary data.

The templates may be used to define any possible kind of model assembly. The model assembly can be as simple as merging some includes to a complete model up to preparing a commit for an external database or providing the whole infrastructure for an optimization. In the case of an optimization this means that LoCo will create, based on the template, not only the input file for the optimization software but also all the necessary infrastructure as for example submit scripts, cleaning scripts, the necessary files for automatic evaluation, configuration files for all involved software components and the whole directory structure. For example the problem with optimization is, that different departments or even users use different responses and constraints and maybe even different optimization strategies. In order to empower them to adapt LoCo to their needs, the standard templates for setting up an assembly for an optimization can be modified by each user. The same approach can be taken for arbitrary solver input. The advantage is that the user can do this using the input formats they are used to. An example of this can be seen in Figure 5. The template language provides not only simple placeholders but also loops and conditions. In fact the template language can be used almost like a simple programming language.

```

$-----
$ DESIGN VARIABLES
$-----
variables [% n_variables %]
[% FOREACH variable IN variables %]
  Variable ' [% variable % ]' 0
  Lower bound variable ' [% variable % ]' -1
  Upper bound variable ' [% variable % ]' 1
[% END %]

```

Figure 5: design variable section from template for LS-OPT input

## 2.4 Team synchronization

Team synchronization is one of the most important tasks to be performed by LoCo. Especially in the case that users from more than one team want to access the same include data it is necessary to establish clear rules for synchronization and enforce them.

LoCo uses the same principles for synchronization as revision control systems that have been in use in software development for quite some time. Much of the synchronization process in LoCo has been inspired by subversion [8]. However the requirements from the CAE engineers are somewhat different than the requirements from software engineers. Requirements are:

- users must be able to work offline
- opening includes in editors and preprocessors has to be as fast as from the local file system
- people have to be able to work even after failure of the server infrastructure
- synchronization has to be controlled by the user
- the versioning has to be controlled by the user
- data has to be transferred in a bandwidth optimized way
- no user should suffer from the errors made by other (files are to be checked before synchronization)
- the synchronization process has to be with good performance and intuitive
- the history of each change in whatever file has to be documented and visualized in an intuitive way
- if possible existing infrastructure has to be used and the requirements in general for the infrastructure have to be kept as low as possible

Whereas some requirements are similar to such solved by revision control systems for software engineering, others are very unique to CAE engineers. Therefore it was not possible to employ a full-grown system as for example subversion which would have been the preferred solution. Figure 6 shows a brief overview of the concepts used by LoCo.

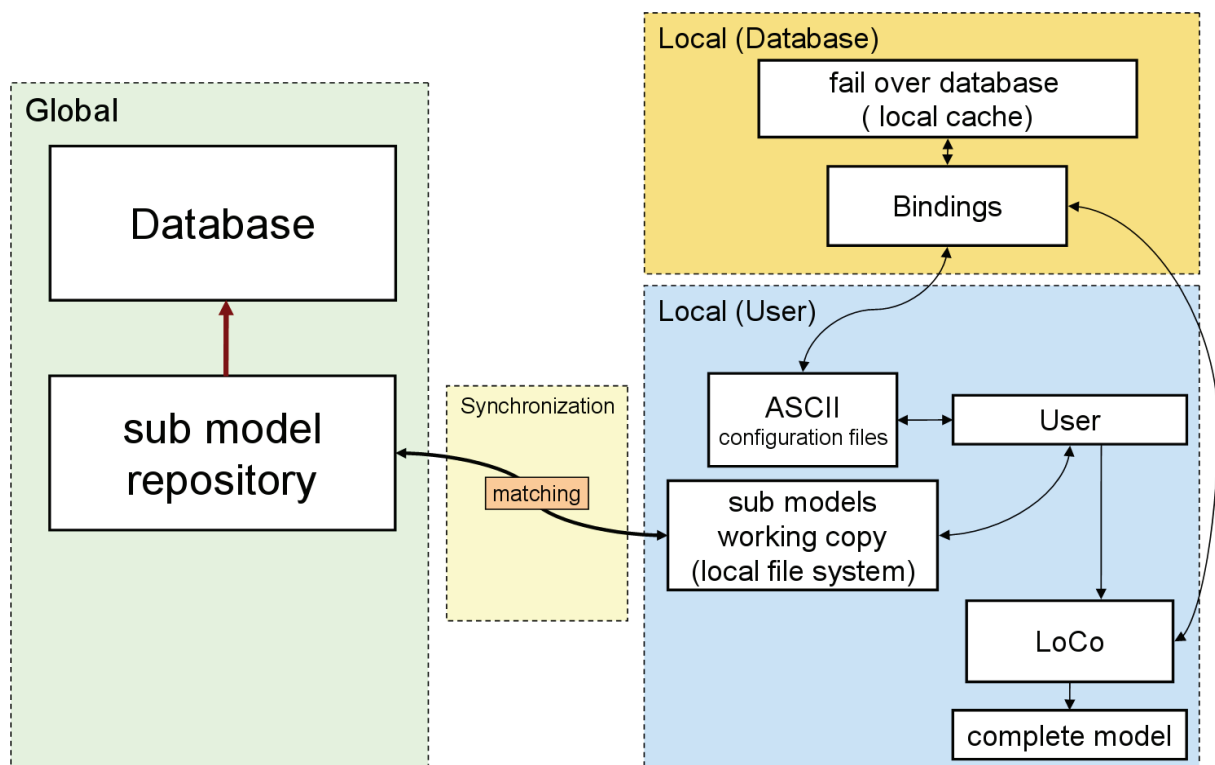


Figure 6: basic client server architecture of LoCo

A global repository has to be synchronized with a local working copy. Furthermore a local database assures that certain operations are performed with good performance. The local database is a cache of the data in the repository. Furthermore the data is presented to the user in form of ASCII files.

However these ASCII files are thought to be just another form of representation of the data to the user. This was necessary because some CAE engineers still prefer the vi or emacs instead of a GUI. Input errors from the ASCII files as well as from the GUI will lead to errors and LoCo will not assemble nor synchronize the data with the global repository. The only difference is that within the GUI we are able to intercept the errors instantly and lead the user through the process of correcting the errors whereas if the ASCII files are used, the users have to fix the errors by themselves.

## 2.5 Integrated optimizations and parameter studies

Parameter studies and optimizations are integrated into LoCo as just another assembly template. From this template the whole infrastructure for LS-OPT is created by LoCo. The user only has to specify which variables he wants to consider for optimization and what load cases he is interested in. The process itself is outlined in Figure 7. The idea is that LoCo creates the infrastructure for LS-OPT but LS-OPT then uses a parameterized configuration file from LoCo in order to employ LoCo to assemble the models.

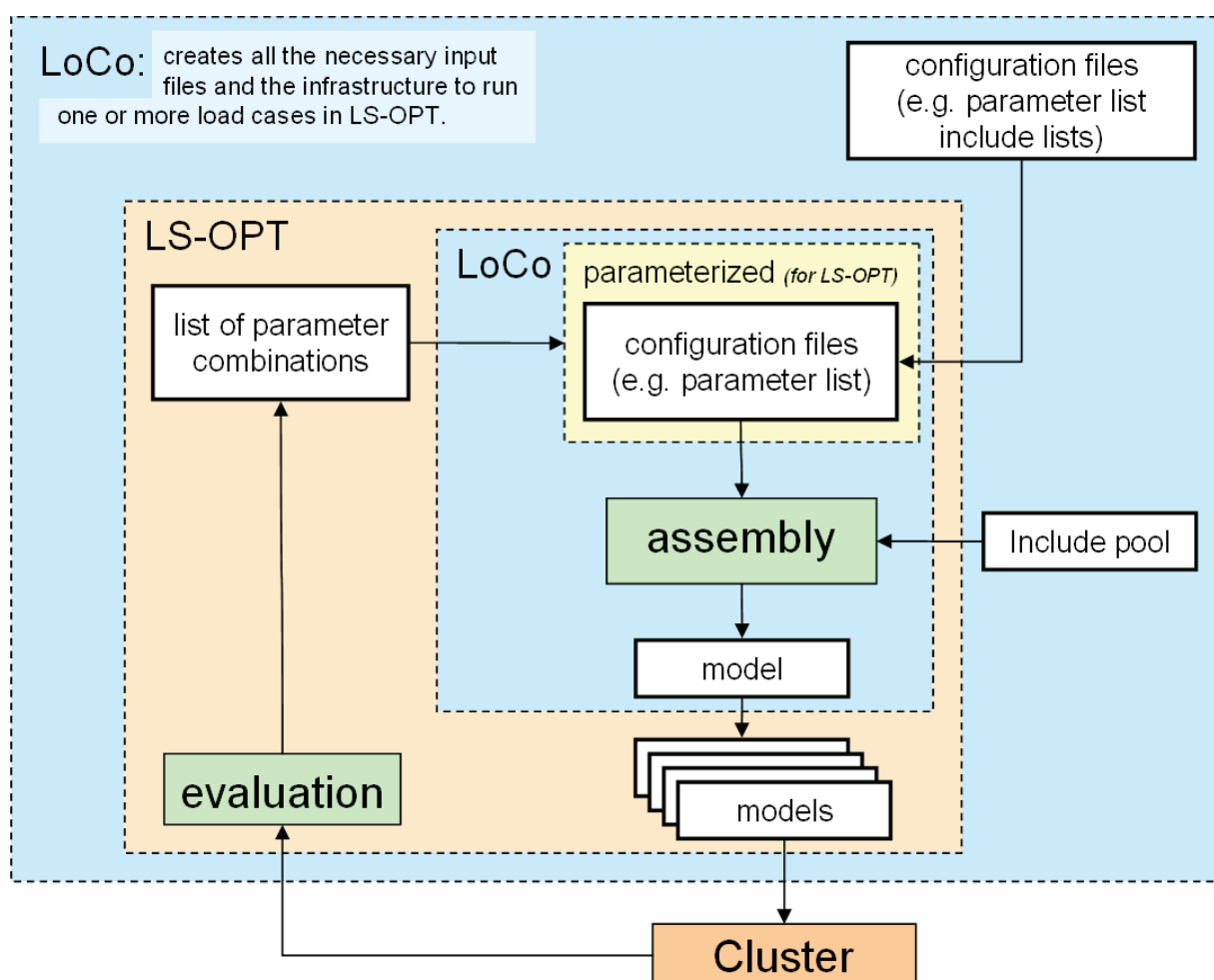


Figure 7: LS-OPT integration in LoCo

The most common form to apply parameterization in LoCo is to use placeholders in text files that are replaced during the assembly process. LoCo searches for that placeholders and substitutes them with a value from the parameter list. However for some cases this is not sufficient. For example the investigation of the effects of geometric changes requires very complex changes of the model that can not be represented by just some simple parameters within the input file. In order to be able to handle such situations, it is possible to provide user defined scripts. These scripts may be parameterized by themselves and can manipulate includes in an arbitrary complex manner or create further includes. This manipulations can range from creating a simple curve for a material, over geometric shape manipulations using morphing tools like ANSA up to very complex pre-calculations e.g. for dummy positioning or airbag folding.

### 3 Parameterization of the width of upper part of a door trim

For the purpose of demonstrating the use of LoCo in conjunction with the parameterization using user defined scripts we will employ an example where the geometry of the door trim had to be analyzed with respect to some injury criteria.

The door trim is shown in Figure 8. The width of the upper part had to be parameterized by morphing. After the morphing the door was used in a full vehicle side impact crash simulation. For the purpose of this investigation only one load case had to be evaluated. However the whole process had to be prepared in a way that allows for adding other load cases. This can be accomplished if the parameterization of the door trim has been prepared for LoCo. Once a parameterized include has been added to LoCo this include can be used for all load cases that work with this include.

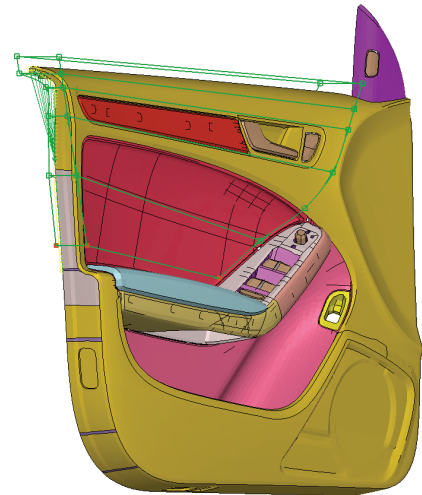


Figure 8: door trim AUDI A4

#### 3.1 Morphing in ANSA

Morphing in ANSA has become a very powerful and easy to use tool for optimization. Furthermore the setup of the optimization process with LS-OPT is now well integrated. Therefore it is possible to set up optimizations with little effort.

The morphing boxes are defined as exemplarily shown in Figure 8 and Figure 9.

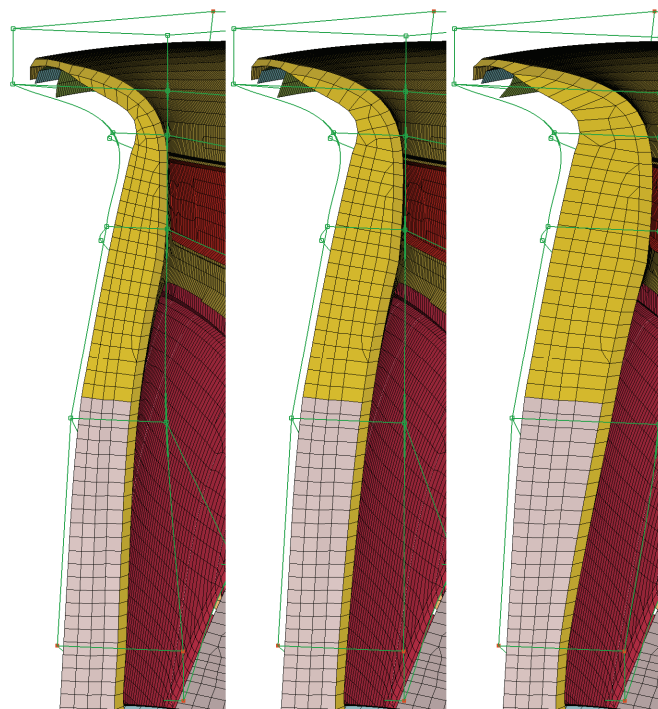


Figure 9: range of applied morphing parameter

The morphing is controlled by check points that can be moved freely or by using different patterns like moving it along edges and others. This “degrees of freedom” can be assigned to parameters which will be written from ANSA into a parameter file. When using the LS-OPT interface of ANSA this parameter file will be prepared so that LS-OPT can read it and pass it with the parameters applied back to ANSA. In order to use this file with LoCo it was necessary to apply the parameterization manually and to define a script that passes this parameter file together with the ANSA model database file to ANSA. Apart from the morphing boxes there is also a task defined within the ANSA model database file. The task manager in ANSA (Figure 10) allows preparing multiple steps that have to be performed when called from a script. These tasks can be everything from applying a simple parameter up to remeshing a whole model. In the past especially the reconstruct feature from ANSA has proven to be a valuable



tool for morphing. The reconstruct function allows to convert a FE model or just parts of it back to topology data and to apply meshing on it in a single step. This is particularly useful in cases where the morphing distorts the mesh up to a point where the element quality is not acceptable anymore.

When reading a PAM-Crash include into ANSA and afterwards writing it back to a PAM-Crash include it may occur that ANSA misses the one or other part of the input. Especially if there were some parts in the include that could not be interpreted by ANSA this may occur. Therefore the original include was provided to the script with all nodes erased. Then after the ANSA model database file had been processed the nodes have been extracted from the ANSA output and merged to the original input file.

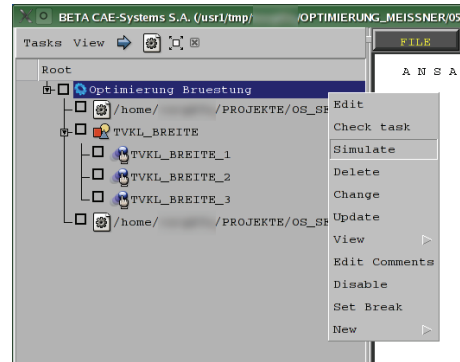


Figure 10: ANSA task manager

### 3.2 Integration in LoCo

Once the process is working all the involved files, such as the original include without the nodes, the ANSA model database file with the morphing boxes, the script, the ANSA parameter file are packed together in a container file which then may be used in LoCo in the same way as original includes. LoCo knows that on encountering such a container file it has to unwrap it and to run the script. This process is taking place each time when LoCo has to assemble a model that includes this container file. The advantage is that the normal user is not all the time confronted with the file overhead of this kind of parameterizations.

### 3.3 Using LoCo to set up the Optimization / Parameter study

Includes that have been parameterized in LoCo may be used in all load cases that they apply for. For that reason LoCo provides the means of setting up highly complex parameter studies or optimizations that involve multiple load cases. This kind of multi load case optimizations would otherwise require a significant amount of time to set them up.

In LoCo the setup of an optimization or parameter study is reduced to providing for each load case that has to be taken into account, the parameters that are to be investigated. Then upon assembly the run mode "OPTIMIZATION" has to be used which will create all the necessary files for LS-OPT

### 3.4 Results

The results of the parameter study of the above example can be viewed with D-SPEX (Design SPace EXplorer). D-SPEX [4] is a software that has been developed by DYNAMore GmbH in cooperation with AUDI. Its focus is the multidimensional visualization of response surfaces. Response surfaces have the advantage that they give a continuous representation of the relation between parameters and responses within the design space.

Figure 11 shows the results of the above example. The result for the injury criteria is shown by means of the width of the upper part of the door trim and the vent area of an airbag. Furthermore the results of all simulations are represented by points.

This kind of visualization can help to understand the system behavior. For example in this case it can be shown that for a smaller door trim the effect of the vent area is bigger than for a broader door trim.

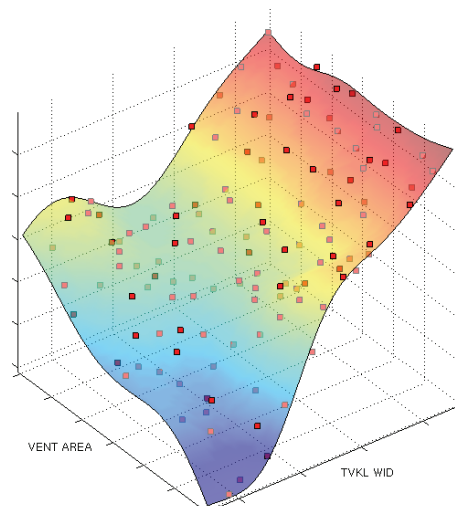


Figure 11: results of parameter study

## 4 Conclusions

With the morphing technology of the pre-processor ANSA shape optimization based on FE-models can be applied. The integration of complex geometrical parameterization with morphing in a highly standardized process like the work flow of LoCo is possible and gives the possibility to apply the parameterized include to various different load cases and scenarios. This together with the deep integration of an optimization tool like LS-OPT empowers the users to use parameter studies and optimization as a standard tool for finding an optimal solution or in order to better understand the model behavior. The DOE-post-processor D-SPEX, which has been developed in cooperation with AUDI, allows a very detailed evaluation of relation between model parameters and system responses.

Usually the main obstacle for applying optimization and DOE-studies in the daily work flow of a simulation engineer is the time consuming set-up of the process flow and the laborious specification of the parameters in the input data. With LoCo these tasks can be completed with a significantly reduced expenditure of time. Thereby, practical application of design studies becomes feasible. For example, the set-up of 10 different load cases, each with a DOE-study consisting of 100 simulations might be performed instantaneous.

Tools for simulation data management and process integrations as LoCo may require a significant effort in development and maintenance. However, there is no alternative to handle the more and more complex requirements in the CAE environment like the increase of load cases, model complexity, number of model series, number of simulations, and steady rise of exchange between CAE-disciplines and project engineers on site as well as with external suppliers and service providers.

## 5 Literature

- [1] Mlekusch B., Dornberg C., Streicher F., "CAE-Prozessintegration und Optimierung", Vortragsband - 7. LS-DYNA Anwenderforum 2008, Bamberg, Germany.
- [2] Müllerschön H., van den Hove M., Mlekusch B., "Optimization Strategies for Highly Non-Linear FE-Applications such as Crashworthiness", Proceedings ECCM-2006, Lisboa, Portugal.
- [3] Stander, N., Roux, W., Goel, T., Eggleston, T., Craig, K., LS-OPT User's Manual, Version 3.3, Livermore Software Technology Corporation, Livermore, 2008.
- [4] Introduction to D-SPEX, Version 0.3, DYNAmore GmbH, Stuttgart, 2008.
- [5] Makropoulou, I. "Template driven LS-DYNA Model Build-up with ANSA Task-Manager", 6. LS-DYNA Anwenderforum 2007, Frankenthal, Germany.
- [6] Uli Göhner, Heiner Müllerschön: "Process Management for Generation and Optimization of Full Vehicle Models for Crash Simulation", 2. Grazer Symposium Virtuelles Fahrzeug, Austria 2009
- [7] <http://template-toolkit.org/>
- [8] <http://subversion.tigris.org/>