Enhancing collaboration on multibody dynamics simulations with simulation data management

K. Schäbe, M. Thiele, C. Zacharias-Tietze
(SCALE GmbH)

Abstract

Today, virtual product development is essential in vehicle projects in large automotive groups in order to keep costs down. As part of the CAE development process, the vehicle's behavior in operational mode must be investigated. It has become common practice to employ multibody dynamics simulations for this task, where the vehicle is treated as composed of various rigid or elastic bodies that can undergo translational or rotational displacement. Multibody dynamics simulation experts then first build up a virtual vehicle model and subsequently let it drive certain maneuvers on various kinds of standardized roads under different standardized driving conditions that apply for all projects.

Managing, sharing and collaborating on the related CAE data, CAE methods and CAE processes across a larger number of simulation engineers can, however, become a challenge. It is therefore important to establish means of working together in teams. For this it makes sense to organize simulation data in such a way that common CAE data files as well as certain process-scripts or simulation-methods in general, can be shared with the whole team. To this end, libraries of common CAE files - such as templates, technical components or connectors that define the vehicle, or the roads and driving conditions that are applied to the vehicle - and all types of process-scripts are created and maintained by a small number of experts. They can then be used by all CAE users working on the different car projects and hence lead to significant democratization of CAE. To ensure that this also works regardless of the location of the CAE engineers, Simulation Data Management Systems such as SCALE.sdm are required to exchange data and allow for effective collaboration. For certain disciplines of the virtual product development process, e.g. handling crash simulations, simulation data management has successfully been introduced many years ago and has been in use ever since. Other simulation domains such as multibody dynamics simulations face different kinds of challenges.

In this presentation, we want to demonstrate how to foster collaboration between multibody dynamics simulation experts using simulation data management tools. We will address the virtual product development process – from vehicle model building to running the simulations and the integration

with the preprocessor – and work out concepts that improve effectiveness as well as consistency for those engaged in the workflow.

In particular, we will focus on the challenges that we have been facing when introducing a simulation data management system for multibody dynamics simulations with MSC Adams for one of our customers.

Since in a multibody dynamics simulation with Adams the model is made up of many individual parts that are each stored in an individual file, the data structure to include in the simulation is extensive and rather complex. Plus, that model has to be combined with a certain road, driving condition and maneuver. It is challenging to reproduce that complex data structure in the simulation data management system – however it is key to make full use of the system's advantages and harvest the possibilities of modularization. These allow for better collaboration on big models when working with different users at the same time by reducing the amount of conflicts or allowing to employ multiple load cases on the same data [1]. However, complexity must not be traded for usability and maintainability such that integrating the preprocessor for easily managing the model is also an important aspect to take care of.

In our presentation, with an example of the integration of a multibody dynamics simulation workflow of MSC Adams for our customer, we will show how to achieve a detailed mapping of the internal data structure of Adams with the simulation data management system. At the same time, we will demonstrate how we manage to foster interaction with the system among the simulation engineers and thus enhance collaboration.

Enhancing collaboration on multibody dynamics simulations
with simulation data management

1. Challenges of integrating a multibody dynamics software into an SDM system

Inherently, when using a multibody dynamics approach to simulating the behavior of a vehicle on the road, one has to take care of many different individual bodies that, when joined together, make up the vehicle. In the multibody dynamics software (from now on "MBD software") this is realized in form of a wide spectrum of file types that underlie a fixed hierarchy and must be organized in certain directories. Not rarely, for one vehicle, several input files in the order of 100 are needed. These files must then be mapped to the SDM system and when working on the vehicle model from the SDM system they need to be reconstructed for each user locally. Additionally, when constructing new vehicle models from prior ones, one might want to substitute just a few files, so the SDM system must be able to also provide global databases that store a large number of input files.

After building the model, load cases must be composed out of different components. In the case of an MBD application, this means that the vehicle is put on a certain type of road where it drives a maneuver. Furthermore, technical details such as payload or information about rims can be included. In other disciplines such as e.g. crash it has been established to implement this in SCALE.sdm with the use of so-called attributes. With this prominent feature of the SDM, include files are attributed by the user with respect to certain characteristics, and then the load case is set up as a specific combination of attributes. When integrating MSC Adams with the SDM, the attribute approach may seem time-consuming to the user at first because attributes would need to be placed on hundreds of files, whereas when running the simulation directly from Adams, only the one file describing the whole model (so-called assembly file) must be selected. However, since in our case, we also maintain the assembly file itself in the SDM (contrary to other disciplines, e.g. crash, where the overall model is assembled from the individual components right before running the simulation) we can directly use the assembly file in the process of attribution and inherit the attributes used for the assembly file to all the files it is referencing. Therefore, we can combine the flexibility of the attribute approach from SCALE.sdm with the compact way simulations are set up in Adams

2. Building the vehicle model

As an initial step when working on a new vehicle model a new project must be defined. In SCALE.sdm, for a vehicle project it is customary to create a so-called pool that contains all the related data in a user-defined directory structure. Then new projects can easily be created from previous ones by substituting data files and making small adjustments to the structure of the pool, if needed. Inside the pool, the current working state of the project is saved in so-called pool versions. The user can create new pool versions for a

new development stage of the project, and at the same time can browse through previous pool versions and thus older project stages. Similarly, individual directories, parameter tables containing values for parametrizing simulations, and even individual include files can be versioned. This approach gives a large amount of control to the user, to tailor the way the include files are presented in the pool to their own needs.

Now, when integrating the MBD software with the SDM, as mentioned in the previous chapter, it is key to reproduce the file structure that the MBD software requires in the pool. Each directory – representing different parts of the whole model - is individually displayed with its own version and can be worked on separately. With this approach, the user can immediately see which files are stored in which MBD software directory as they would on their local machine, and when working on a vehicle model, the whole file structure can easily be reproduced on a machine and fed to the MBD software. Therefore, the main directory created in the pool will contain that include structure.

A large amount of other files possibly needed for the model are stored in so-called subpools. These pools can be reached and maintained individually like every other pool but can be linked to the file structure of various main pools. Therefore, a subpool can be used as a library or a global database that offers files for the main vehicle project and shares them among different projects. When integrating the MBD software, in subpools, it is feasible to store files there that contain smaller technical elements that will be used in the different vehicle projects. Additionally, files containing information on roads and maneuvers can also be stored in a separate subpool. These are not needed for creating the vehicle model, but when setting up load cases later. Such, each vehicle model can be combined with every road and maneuver. These are standardized for the whole organization and maintained exclusively by experts who have special permissions and by this provide their expert knowledge to the other team members.

For working on the vehicle model from the SDM, so-called external apps can be deployed directly through the SDM system. These applications are customary scripts that allow specific operations on the data stored in the pool and can be tailored directly to the user. The external apps too, are developed and maintained by a few experts and are then available for all users. For the integration of the MBD software MSC Adams into the SCALE.sdm, an external app was developed that allows to open the model with Adams directly from the SDM system. The user can then work on the model from their local workstation, e.g. make adjustments or create new files, and then after saving the project, the external app automatically updates files that were changed in the SDM and imports new ones. The app also takes care of reproducing the file structure the MBD software requires locally for the user. With another external app, the user can easily import complete data structures from their local

machine into the SDM without having to import each file individually which saves a great amount of time and reduces possible errors.

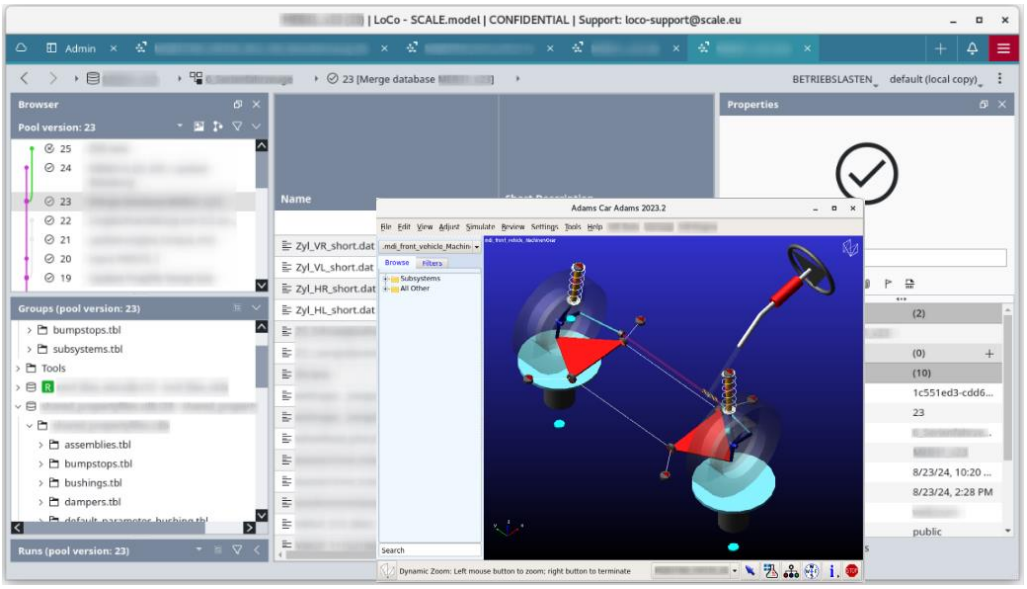

Figure 1: Part of a vehicle model, displayed in MSC Adams (small window), opened directly from the SDM System with (large window).

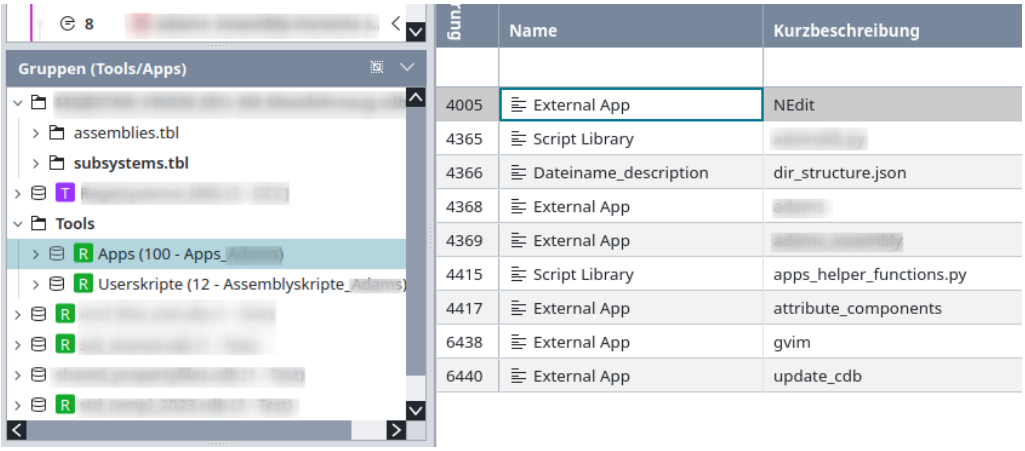

Figure 2: Detailed display of the directory structure of the SDM, with the Apps-subpool highlighted in blue (left). Apps stored in the Apps-subpool (right). The green R-tag makes clear that this pool version is the reference version that all users should use for their work.

3. Running the simulations

Once the design process is finished and the vehicle model exists, load cases must be set up. As described previously, load cases are defined as a

combination of attributes put on the include files, plus a set of parameters if one wants to explore the effect of a certain variable and therefore parametrize it.

In the discipline of multibody dynamics, we want to test the vehicle under certain driving conditions. Therefore, as the initial step, a road and maneuver must be chosen. Furthermore, details on technical elements can be given, e.g. the hubload or the type or size of a rim. With all these properties set, the files needed for the simulation will be exported from the SDM to a machine (the user can decide whether to run the simulation locally or on an HPC) and the simulation will be run. In another subpool, to set up and run the simulation, additional process scripts are stored that can also be tailored to each user's needs. Again, few experts work on these scripts and then provide the scripts to all users via the subpool which acts as standardized library.

Similarly, as during the process of building the vehicle, also for setting up load cases it is feasible to automize certain steps of the workflow and relieve the user from performing them manually via an external app that is stored in the apps pool. In this case an external app was developed to automatically attribute those files that are needed for the simulation. The fact that all files relevant for one vehicle model are hierarchically called from the assembly file is exploited here – the user only has to apply attributes to that assembly file manually and afterwards executes the app on it. Such, the same attribute values will be given to all files originating from the assembly file, and thus all files of the vehicle are available for the simulation.

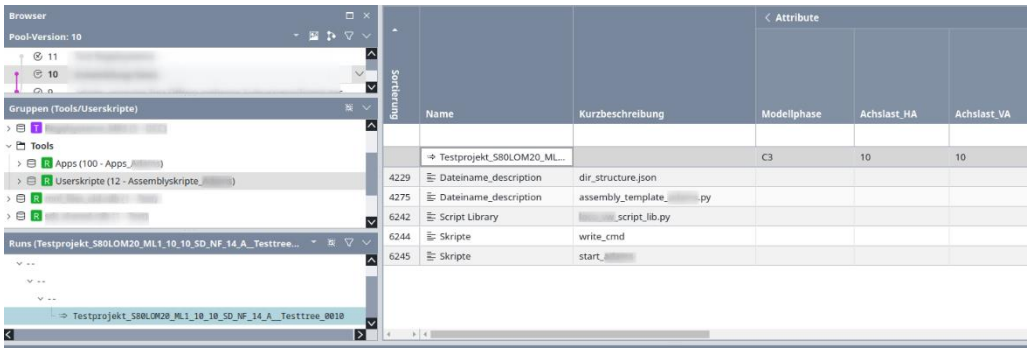

Figure 3: Setup of a simulation run, highlighted in blue. In the name scheme of the run e.g. the attributes used in that run are mentioned. Highlighted in grey is the Userscripts-subpool (reference version R). Process scripts relevant for the simulation run are displayed right, e.g. the script that runs the MBD software from the SDM.

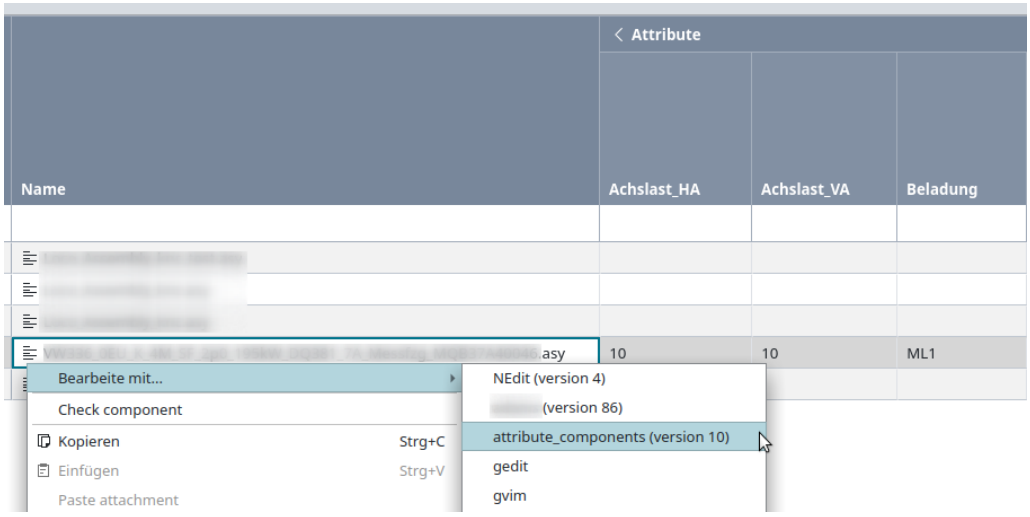

Figure 4: Attributed Assembly-(.asy-)File. By right-clicking on that component the user can choose to execute an external app, here the scripted app "attribute_components" is highlighted in blue. This operation will give the attributes that the *.asy-File has to all subordinate file referenced in that *.asy-File.*

4. Potential adjustments to the workflow and data analysis

After setting up the current workflow, we identified a few possible improvements. So far, first, the vehicle model is built and all the relevant information stored in the assembly file. That file is then used for the simulation. Another, more flexible way could be to start the simulation process from the SDM, i.e. all relevant model files and process scripts are already being exported to the machine where the job is run, but then the user still has the option to work once again on the model from within the MBD software. Adjustments being made in files would be imported backwards to the SDM, and the simulation would be run with the edited files. If needed, also more include files could be requested and exported from the SDM. The assembly file would then be edited or even newly created only after starting the simulation. This workflow would give even more flexibility to the user but requires many include files already being attributed. Therefore, it can make sense to switch only after several simulations have already been run.

As a final step, in SCALE.sdm, it's also possible to directly analyze simulation results (e.g. extracting key values, plotting curves, taking pictures and videos) and import related result files. Such, the user has a complete overview of the whole virtual product development cycle in one place.

5. References

[1] R. Bitsche, M. Thiele, T. Landschoff, and M. Koch, “Recent Developments in LoCo Instant Collaboration in Simulation Data Management,” no. 2, 2017, Accessed: Feb. 03, 2025. [Online]. Available: https://www.scale.eu/wp-content/uploads/2022/04/2017-LS-DYNA-slides_Recent_Developments_in_LoCo_Instant_Collaboration_in_Simulation_Data_Management.pdf