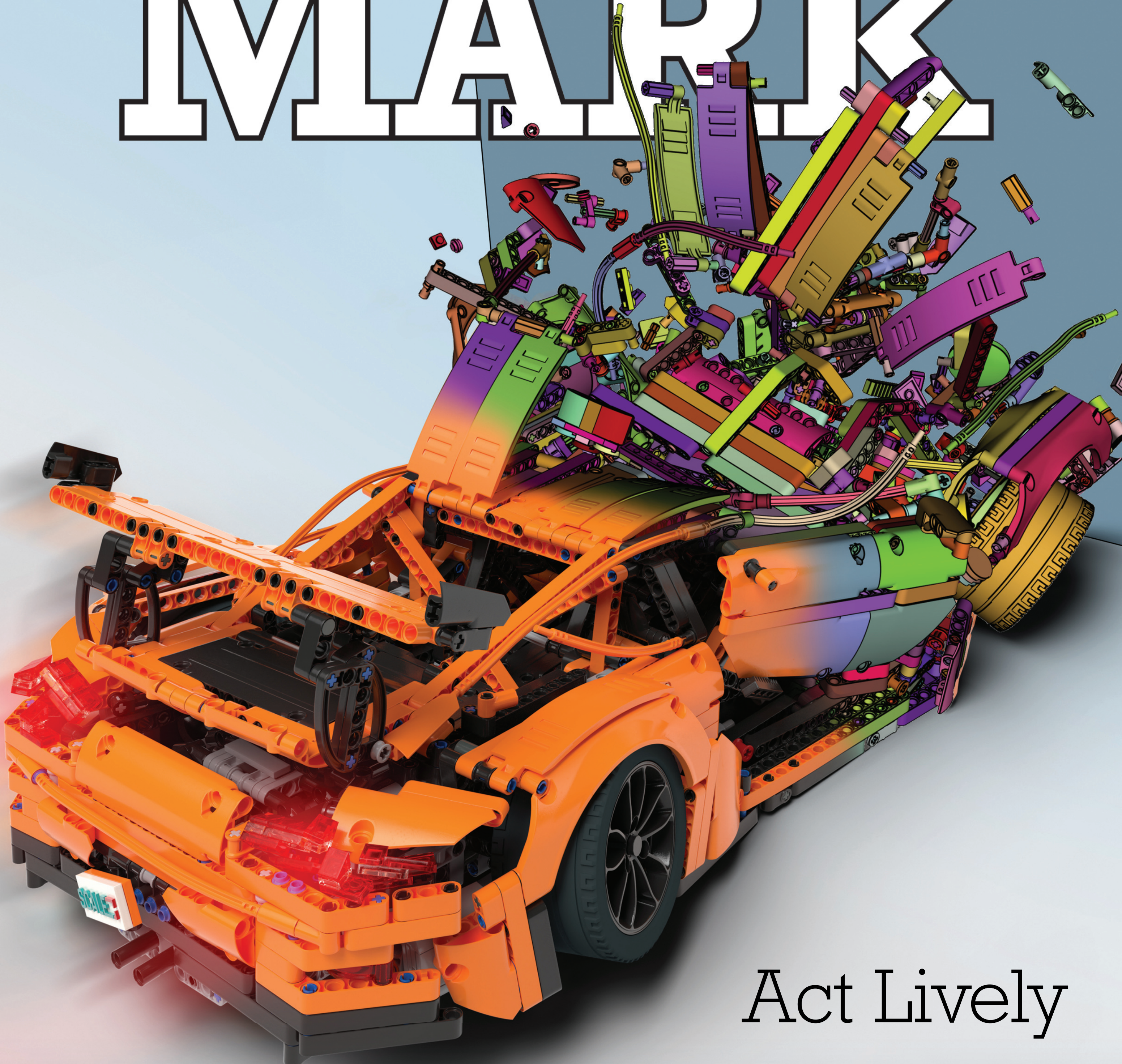
BENCH MARK



Act Lively

NAFEMS

JULY 2023

THE INTERNATIONAL MAGAZINE FOR ENGINEERING DESIGNERS & ANALYSTS FROM NAFEMS

Virtual Product Development with an SDM-System Demonstrated by Playing with LEGO® Models

Marko Thiele, Martin Liebscher & Gordon Geißler | SCALE Gmbh

Given that in our professional lives we are dealing with highly sophisticated crash models on a daily basis, it seems obvious that we instantly thought we should be able to simulate a crash of a LEGO® Technic Porsche Model using an explicit crash solver after seeing a video of a physical crash of this model on YouTube [1]. However, it turns out that setting up a simulation project for such a model with its thousands of parts is quite challenging and needs to be done in a very structured way. Therefore, we decided to use this crash simulation as a benchmark for our simulation data management (SDM) solution.

Setting up a comprehensive continuous CAE-process is a complex task. It usually involves managing the requirements for the desired product, working with CAD data to create a virtual prototype, meshing the geometry for preparing the finite element analysis, dealing with a multitude of sparse CAE solver files to create actual simulation runs, submitting jobs to the HPC or cloud for solving and subsequent monitoring of the simulation runs, handling the result files, deriving key-results, and finally creating reports for the simulations. For real-world car development projects, where hundreds of experts must work and collaborate in such a process, this can be an overwhelming task.

To showcase the basic principles of how such a workflow could look, we created an example using LEGO® car models.

Facilitating the CAE Process by Using SDM Systems

When it comes to SDM, among the many aspects of the CAE process that must be covered, one of the most important is the integration of CAD data, either by integrating a PDM system or importing the CAD data for meshing and organizing the teamwork for this process. The ultimate goal of this would be to have a CAE process that can directly integrate with the PDM system and work fully automatized directly on the CAD data. However, in currently ongoing simulation projects of automotive crash simulations, the process of getting from a CAD model to a simulation model still involves a lot of manual steps.

For example, when we initially tried to download and mesh the CAD data for the Porsche model, we found that dealing with such a huge number of parts in a model is rather challenging. A particular problem, the fact that the same bricks were repeatedly used at multiple locations in the model, made it obvious that we needed a more structured approach to handling the simulation models. Starting with small models we built up a process within our SDM system where we used a library of meshed parts which can be automatically assembled to a complete FEM model directly from the CAD data. In the end, by using this approach, we were able to assemble the whole LEGO® Technic Porsche model which consists of 2704 individual bricks and a total of 18 million elements.

The simulation process and setup of the simulation models which use the LS-DYNA® solver has been described in detail in our publication from the 12th European LS-DYNA conference [2].

Teamwork

When we started setting up the simulation process for generating LEGO® models made from thousands of bricks, it became clear that this ambitious task would require teamwork. Because we set up the whole process in our SDM system, we can showcase how we addressed this. Through the SDM system colleagues from SCALE GmbH in the Ingolstadt and Dresden offices, DYNAmore in Stuttgart, as well as colleagues in France and the USA were able to collaborate. The SDM system made it possible to set up the simulation processes, handle the data required for simulation, and share that data among various team members.

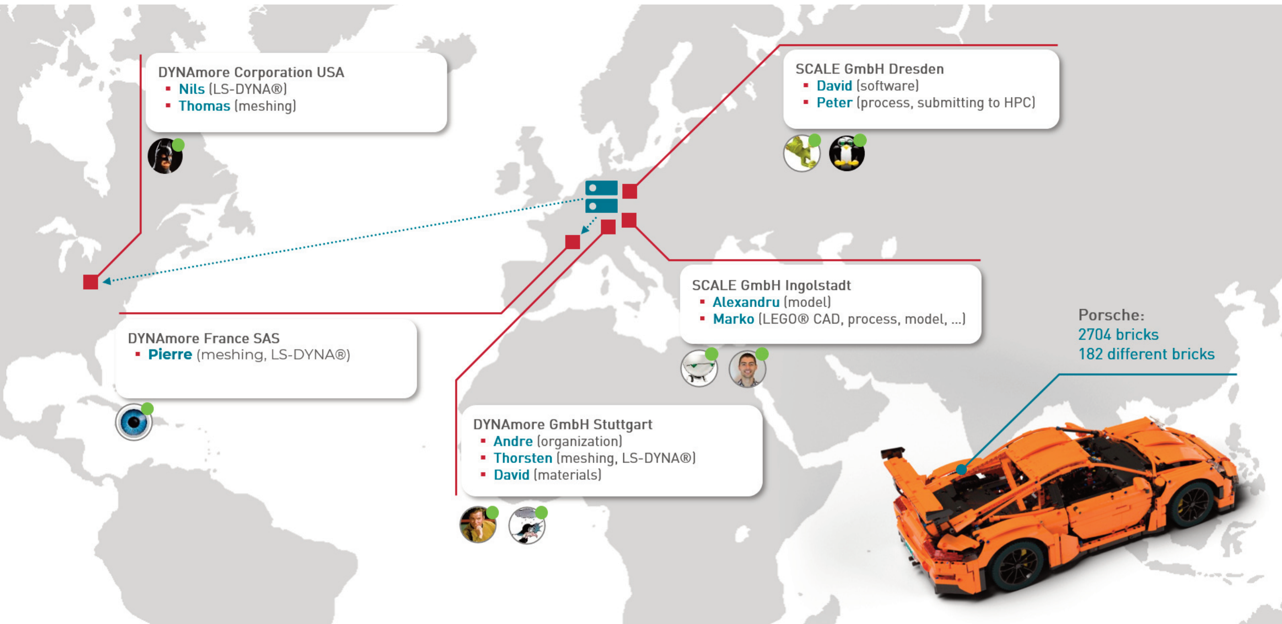


Figure 1: Collaboration during the set-up of the LEGO simulation within the SDM system.

The main challenge when working together on the same model is that no more than one person can work on the same pieces of data at the same time. The problem at hand is that the typical CAD and CAE preprocessing tools are working on files. To collaborate on the same CAD or CAE model, a commonly used approach is to split up the model into several files. As the size of the team grows, the importance of splitting up the model into ever smaller parts increases. This is also something that is seen in productive examples of real car simulation projects and is especially challenging with regard to the user interface as well as the underlying mechanisms, such as synchronization of data.

CAD Data

The most widely used file format to describe LEGO® models is LDraw [3]. In this format basically every line is just providing the transformation information, a color, and a reference to one brick in the standardized brick library. When trying to abstract from the LDraw format to actual LS-DYNA® simulation models it seemed natural to use a similar approach as used in the LDraw format.

The goal is to be able to import these models relatively easily and use the positional information of the bricks to position the actual meshed bricks. This way each brick only has to be meshed once and can be used instantly wherever it appears in the model. The idea is that, if a library of meshed bricks in the same positions as the bricks in the LDraw library is available, it should be easy to create simulation models for any of the countless models available online. A meshed brick can be used over and over again by utilizing the solver's *INCLUDE_TRANSFORM cards to import the brick to its various locations. Within the SDM system the meshed bricks are maintained in a shared library pool that can be used throughout various projects (Figure 2).

Integration of CAD/CAE Preprocessors

Although storing the sub-assemblies separately in our SDM system is paramount for working simultaneously on different parts of the model, this approach turns out to be challenging when working with CAD/CAE preprocessors. In order to edit the complete car model in the preprocessor, special measures are necessary to combine the sub-assemblies in one file or file structure and open it with the required software (for example Primer, ANSA, LS-PrePost, Hypermesh, …). This is a very common problem when integrating preprocessors with an SDM system where the components of a simulation model must be managed in a modular way.

In our case, we set up the integration of a preprocessor using LeoCAD [4]. The process of using LeoCAD directly from the desktop client of the SDM system is implemented in a python script. This script parses the sub-assemblies and combines them in one file.

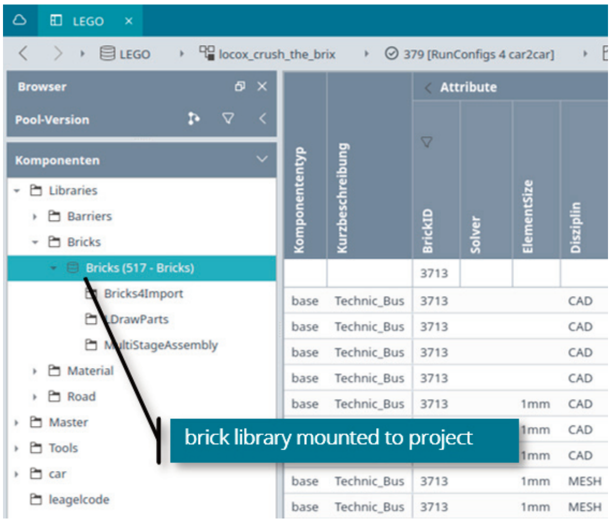


Figure 2: Brick library with CAD and CAE data of meshed LEGO® bricks.

Once opened, the user can work in the GUI (Figure 3) and edit the model. After saving the changes, the model is disassembled, and the individual sub-parts are updated automatically in the SDM system.

Meshing

The main collaborative effort relates to the actual meshing. This has been done by colleagues from DYNAmore GmbH, DYNAmore Corporation in USA, DYNAmore France SAS and SCALE GmbH using the SDM system to organize the teamwork and exchange the meshed bricks (Figure 1). Within the SDM system, all involved data can be managed in one place. The source CAD files of the bricks are kept together with the cleaned geometry and raw meshes, and are stored as ANSA [5] DB files as well as the LS-DYNA® models.

By keeping a record of the meshed files and the sources of CAD data they relate to, it is always clear which CAD version the meshed part has been derived from in the first place. This is also a very common problem in the industry when trying to achieve consistency in handling CAD and CAE data.

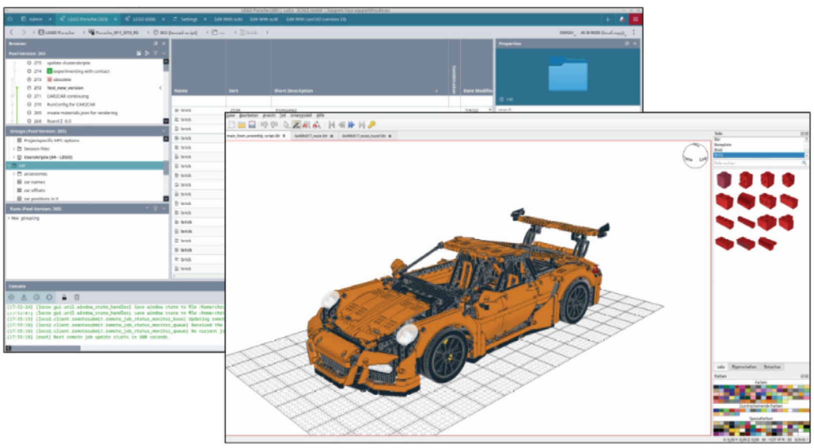


Figure 3: Working with CAD data within an SDM System.

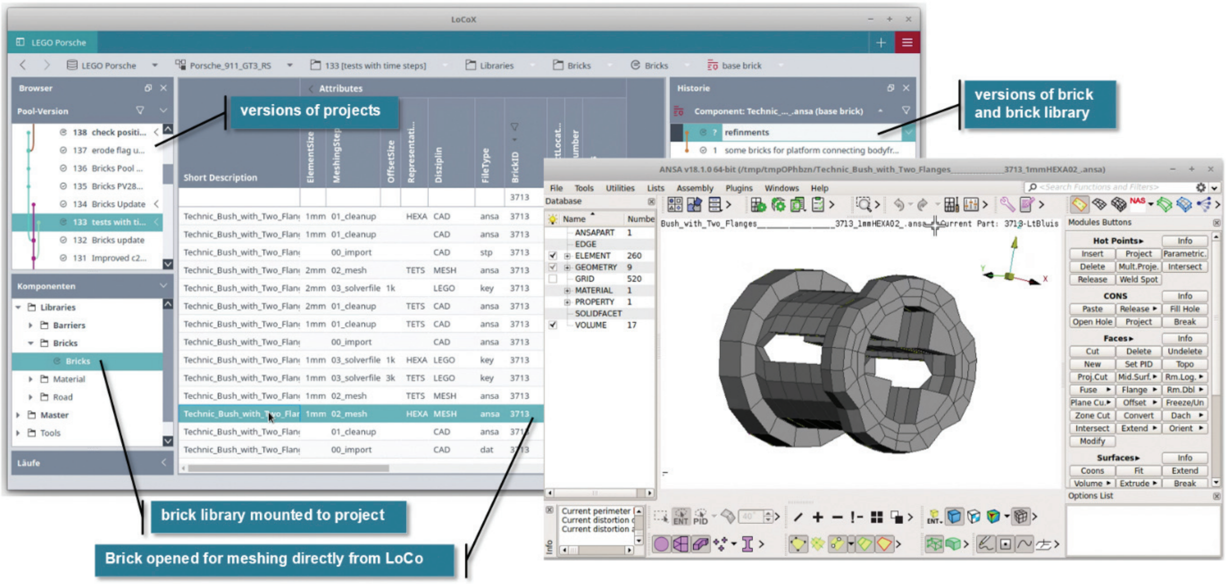


Figure 4: Managing the library of meshed bricks in the SDM system.

Assembly

Other files needed for the final simulation, such as a main file with all the control cards, barrier models, guiding blocks, files with material cards, and so on are also maintained in the SDM system. Finally, a script is stored together with all the components of the model, which is executed by the SDM system when assembling an actual solver deck. This script takes the transformation information of each brick and translates it to *INCLUDE_TRANSFORM and *DEFINE_TRANSFORMATION cards such that each *.key file of the bricks is imported over and over again. The same script also creates session files for LS-PREPOST [6], Animator [7], and META [8] for correct coloring when visualizing the resulting crash animations during post-processing. A more detailed description of these principles is given in our earlier publications [2] [9].

With this SDM setup in place, it becomes easy to apply changes to the LEGO® CAD model by using, e.g. LeoCAD or importing other LEGO® models and running these as new simulations. Provided the meshed bricks already exist for the bricks used in the model, the simulations can be run right away.

Apart from the big LEGO® car models, we also created and uploaded various smaller models. Given the much smaller number of elements, these simulations run much faster, which allows for a wider variety of simulation setups and disciplines, which makes them a lot of fun to play with.

Load Cases and Variants

Once the entire simulation process and a running simulation model has been set up in the SDM system, it becomes easy to create different load cases since many aspects of the model can be parametrized. Besides the 40% ODB offset barrier crash shown on YouTube [10] [11], load cases for frontal rigid wall, 50% and 25% offset barrier, and crashes with 30° impact angle and various velocities were set up as well.

Our SDM system has the ability to use attributes defined on components and parameter values in order to relate them automatically to specific simulations. This makes it possible to create a high number of different variations on one virtual product without having to relate all components (files, includes) individually to each load case. Therefore, a change applied to, e.g. one brick, is automatically taken over to all defined load cases. In this way, submitting the jobs for many load cases to evaluate an overall status of a simulation project becomes effortless.

Figure 5: SCALECAR, Mercedes SLS AMG, AUDI Quattro.

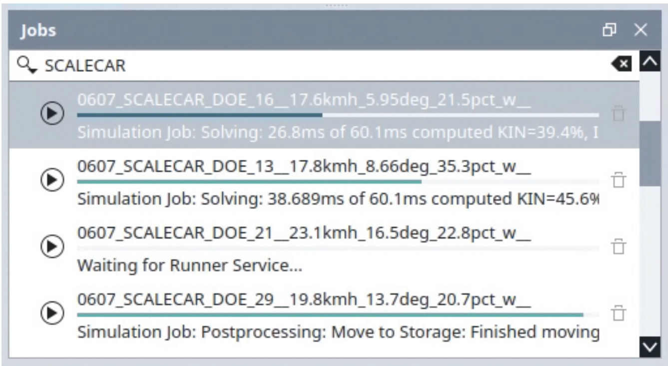


Figure 6: Monitoring of jobs from SDM system

Job Submission

Submitting a simulation job to a high-performance cluster (HPC) can be done right away from the user interface of the SDM system. This is another benefit of having an SDM system. The user does not have to care how the data is going to be copied to the HPC system and how results can be retrieved after a job has been finished. HPC workload can even be spread across multiple different data centres or cloud resources, so the users do not need to know where their simulation jobs are running. The progress of the jobs can also be monitored directly from the user interface of the SDM system (Figure 6) so that the users always know when to expect results.

Simulation Disciplines

One very important aspect of any SDM system is its ability to integrate all kinds of different simulation software and the corresponding pre and post-processing tools. One way of integrating preprocessing tools has been described in a previous section. But there are many other aspects of integrating the various simulation disciplines. Furthermore, even if the disciplines were to use the same solver, each simulation discipline would likely require completely different approaches to handling the data and process.

Apart from the already mentioned crash simulations, we also created a process for car-to-car simulations, using LS-DYNA® as a solver and CFD simulations using OpenFOAM. And finally, to demonstrate the flexibility of our SDM system with regard to supporting different simulation disciplines and solvers, we also integrated the rendering of high-quality pictures and videos as another simulation discipline.

Car-to-car Simulations

For creating a car-to-car simulation, we used the Multi-Stage-Assembly feature of our SDM system. For this, we adjusted the process such that after the assembly process of the individual cars the CAD data of the positioned cars are automatically re-imported back into the SDM system. These assembled and positioned cars are parameterized and can thus be placed automatically in a car-to-car crash scenario and used for the actual simulation.

The final car-to-car model for the Porsche vs. Bugatti load case was composed of 6303 LEGO® bricks resulting in ~45 million volume elements. With these big solver files, even just opening them as full models in tools such as Primer or ANSA is problematic. It can still be done, but it takes a considerable amount of time and needs quite some memory.

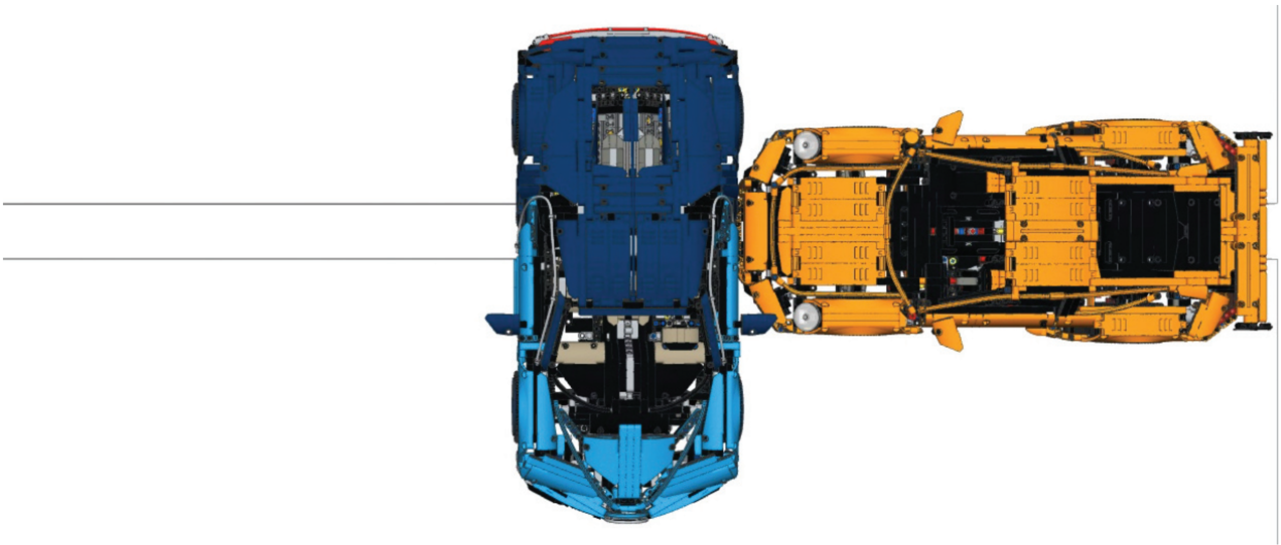


Figure 7: Final car-to-car crash set up. Porschedriving at 60km/h hitting the Bugatti on the left.

Figure 8: Rendering of LEGO® Porsche FEM model.

Rendering

The rendering of photo realistic images to create a physically correct visual representation of the product can also be seen as a simulation of the behaviour of light and material. The solver for the discipline is the open-source software Blender [12] which is also widely used for creating animation films.

For this purpose, the Multi-Stage-Assembly feature has been used to reference the d3plot results files of the simulations as input data within the RENDER project. The solving process is divided into two phases. In the first phase, the d3plot.fz files are converted into Blender files and, using a script within Blender, materials are assigned to each part and post-processing is done for all meshes to achieve acceptable rendering results. In the second phase of the process, Blender is finally used to create the actual rendering.

The submission of rendering jobs to our in-house HPC system can be done in just the same way as with the explicit crash solver jobs; through the user interface of our SDM system.

Computational Fluid Dynamics

One of the latest disciplines that has been added to our LEGO® example in our SDM system is external aerodynamics using Computational Fluid Dynamics (CFD). Running CFD simulations within an SDM system is challenging because the mesh of the model is, other than in FE models, not an addition of smaller meshes of part meshes. Rather than storing meshed parts, for CFD runs the meshing settings in our process are stored and managed, creating slightly different meshes for every model configuration during runtime. An additional challenge is the management of huge CFD meshes and result files.

As a solver, we used OpenFOAM, and just as with the process for LS-DYNA® we used a python script to automatically convert the LEGO® CAD data into *.obj files that could be used for the automatic meshing. With this approach it is possible to start CFD runs right away after changing the LEGO® CAD files.

Similar to the rendering discipline where the conversion of crash results and preprocessing of meshes for rendering can take quite some time, CFD meshing can also take up a considerable amount of time. In these cases, the result of the meshing can also be stored in a temporary place on the HPC system. This way, the meshing only has to be repeated if geometry changes, not for every CFD run.

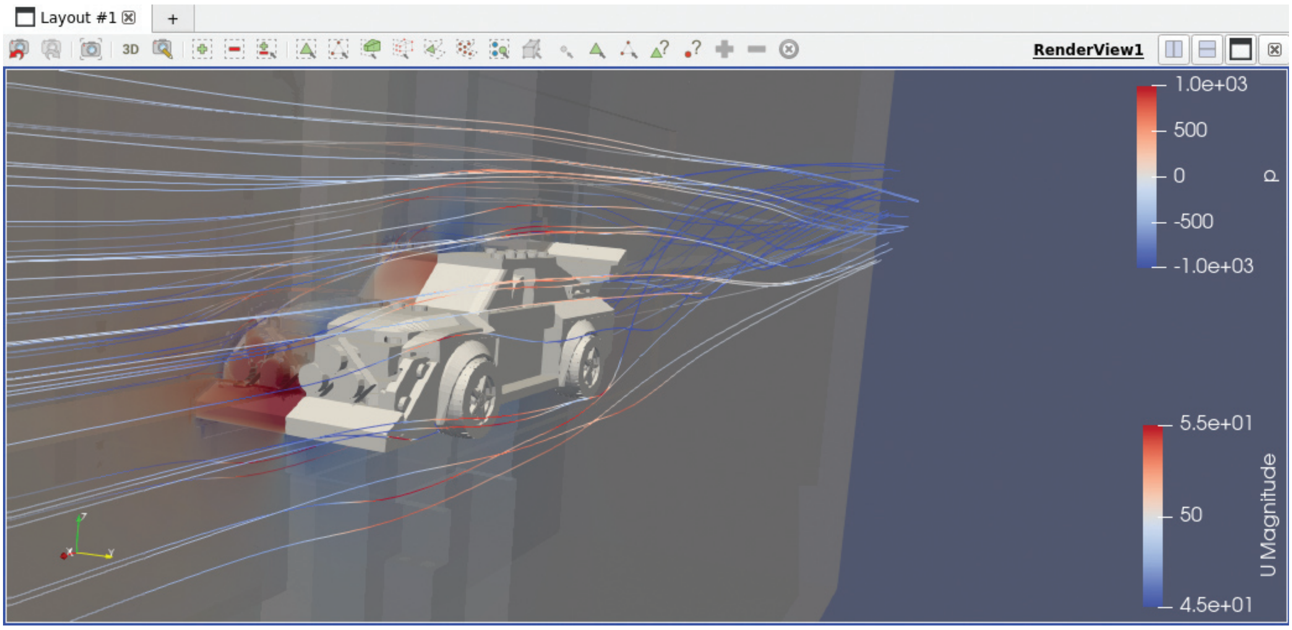


Figure 9: CFD results of AUDI Quattro LEGO model visualized in Paraview.

Post-processing

Being able to do a quick review of the results even when working with a laptop, and without having to download the complete result files and open them in preprocessing tools can save valuable time. To demonstrate how such a setup works within an SDM system, we implemented a process where the results of each simulation are post-processed directly on the HPC system. The simulation and the extracted results are then stored in the SDM system, where they can be reviewed instantly from the web interface of the SDM system after the simulation finishes.

As they are stored within the SDM system, the results are also archived and accessible for all team members working on the project regardless of where they are.

The necessary steps for performing the extraction of data from the results are also set up and defined within the SDM system itself. For example, there are several scripts maintained within special library pools to create images and videos, extract curves, key values, and so on.

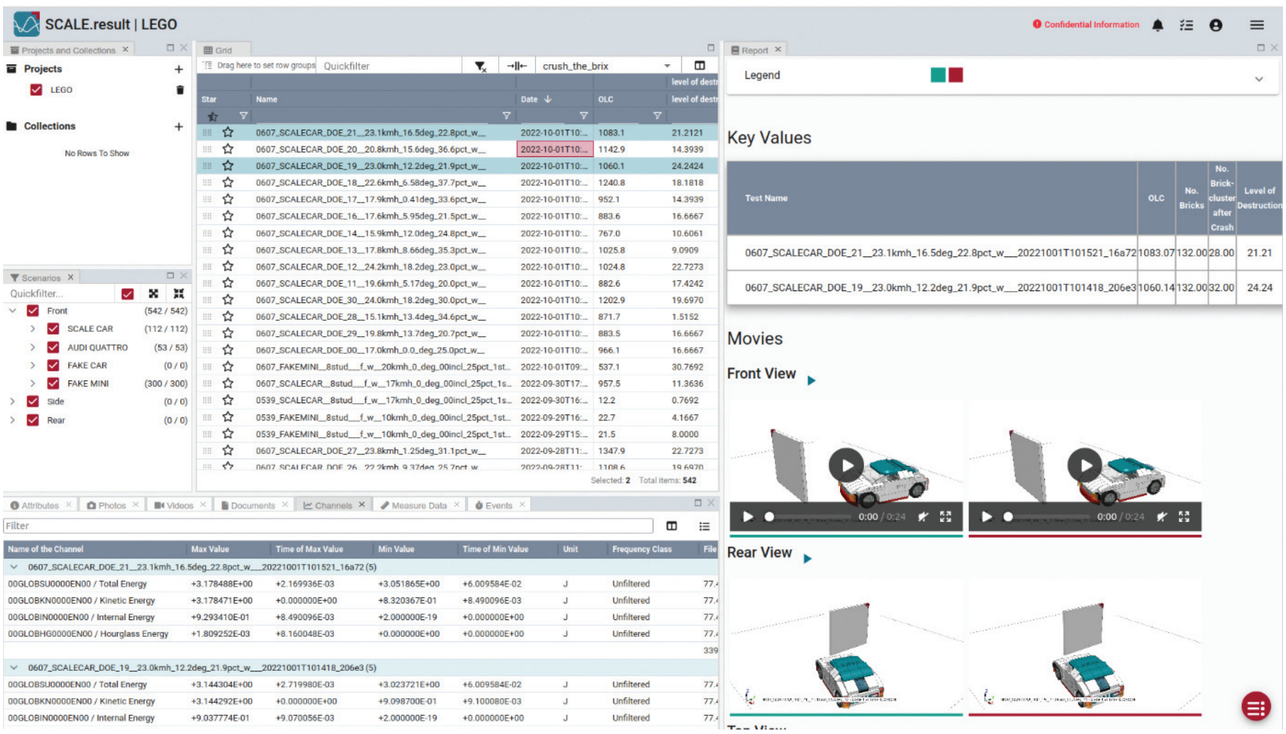


Figure 10: Web interface of SDM system for convenient accessing result data.

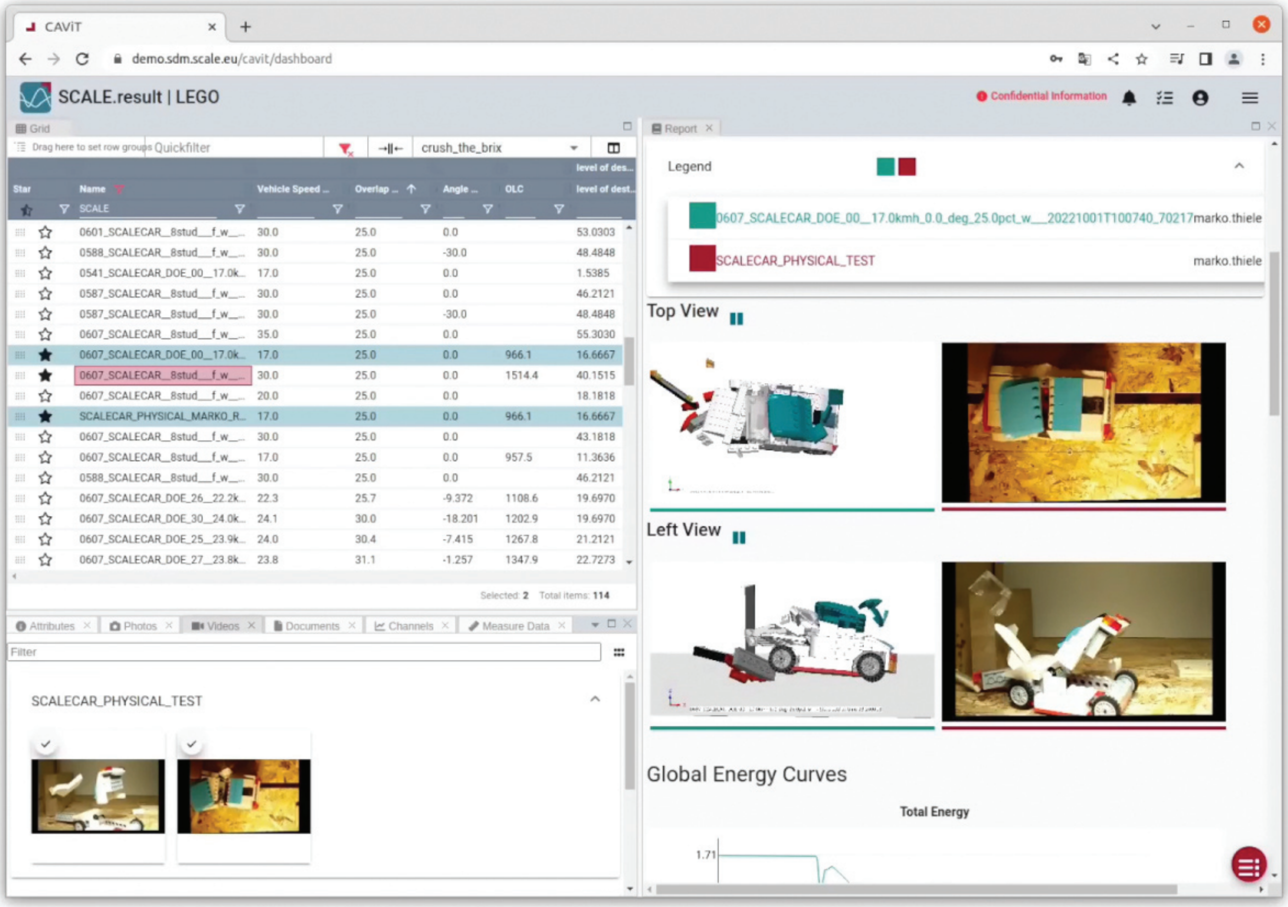


Figure 11: Comparing test and simulation data within SDM system.

Comparison Between Real Tests and Simulation

Other than for the comparison of simulation results with other simulation results, another major use case is the comparison of simulation results with results from physical tests.

For this purpose, it is possible to upload test results into the SDM system using the ISO-MME format. In cases where a dedicated test data management (TDM) system exists, test results can also be fetched automatically from the TDM system and displayed side by side.

For the LEGO® crashes we got physical test footage for the small SCALECAR model which has been filmed at 240fps from the left side as well as from above. The videos have been prepared such that they correspond to the videos automatically created during the post-processing step of the simulation. The test data has then been packed into ISO-MME container and uploaded manually to the SDM system where it now can be used to compare the simulation results with the test data (Figure 11).

Being able to compare the test and simulation results rapidly is very convenient for validating the simulation results.

Data Analysis with Simulation Results on an SDM System

Within the SDM system all simulation results possess a lot of meta data in a very structured form. This makes it not only possible to search for simulation runs with certain properties but also to use methods of data analysis to investigate trends within the simulation results.

Besides enabling the reviewing of existing accumulated data, the parameterized models within the SDM system make it very convenient to create DOEs to study the effects of numerous parameter constellations.

For example, for the LEGO® crash models, we have parameterized the vehicle velocity, the angle of impact, and the overlap to the side with the barrier. Using the Multi-Run feature of our SDM system [13] we created a parameter table with a DOE for these three parameters.

Once all jobs are terminated and the results are available on the SDM system, the user can start examining them in the web browser by creating scatter plots, sensitivity analyses, or doing outlier detection. They can even create a TensorFlow machine learning model and generate 3D visualizations for these meta models (Figure 12).

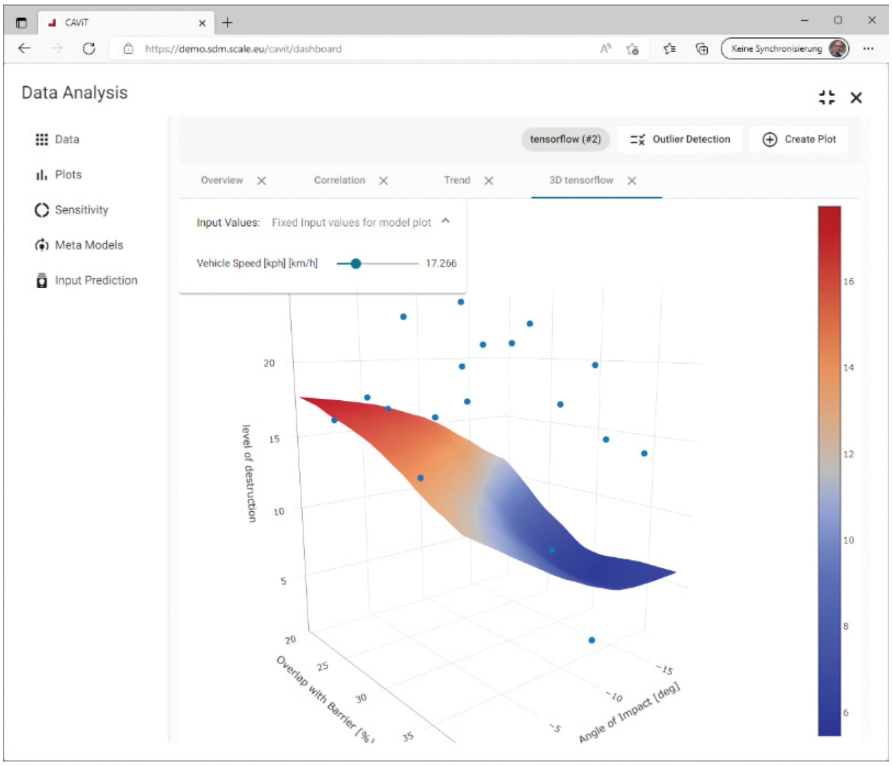


Figure 12: 3D visualization of TensorFlow model for inspecting the "level of destruction" with respect to barrier overlap and angle of impact.

Summary and Outlook

Creating the LEGO® example processes in our SDM system is always a lot of fun because of the teamwork involved. Many of the colleagues who have helped make this possible worked on the project during their spare time or whenever there was some time in between other project work. Solving the problems associated with the process setup for the SDM system and the LS-DYNA® or OpenFOAM models as well as the Blender rendering is quite some pleasure on its own, doing all this with an actual LEGO® model made it even more interesting and enjoyable.

The current setup within our SDM system makes it now relatively easy to set up a simulation model from LEGO® models, and the usability of dealing with the models and the processes is getting easier all the time. The final goal would be for one to be able to download models from the OMR of Ldraw [14], bricklink.com [15] or create an own model, easily define some boundary conditions, and run crash or CFD simulations and create beautiful renderings of the results. ■

**LEGO® is a trademark of the LEGO Group of companies which does not sponsor, authorize, or endorse these investigations.*

References

[1] LEGO® crash tests by ADAC with a Porsche Technic model, https://youtu.be/dCPWPj4JHqg

[2] M. Thiele, T. Gerlinger, D. Koch, A. Haufe, N. Karajan, T. Weckesser, P. Glay, A. Saharnean, “Simulation Data Management from CAD to Results with LoCo and CAViT for Large Scale LS-DYNA® LEGO® Crash Models” presented at the 16th International LS-DYNA® Users Conference, 2020.

[3] LDraw.org, http://www.ldraw.org

[4] L. Zide (1998). LeoCAD (Version 18.02) [Software], https://www.leocad.org

[5] BETA CAE Systems (1990). ANSA [Software], https://www.beta-cea.com/ansa.html

[6] Livermore Software Technology Corporation (1987). LS-PREPOST (Version 4.6) [Software], http://lstc.com/products/ls-prepost

[7] Gesellschaft für Numerische Simulation mbH (2004). Animator (Version 2.4.0) [Software], https://gns-mbh.com/products/animator

[8] BETA CAE Systems (1990). META [Software], https://www.beta-cae.com/meta.htm

[9] M. Thiele, T. Gerlinger, D. Koch, A. Haufe, N. Karajan, T. Weckesser, P. Glay, A. Saharnean, “On the Setup and Simulation of Large-Scale LEGO® Models built with LS-DYNA® and LoCo” presented at the 12th European LS-DYNA Conference, 2019.

[10] S. Hansen, “Lass krachen! Der Lego-Porsche-Crash“ c’t magazin, issue 12, 2017, pages 74-79

[11] heise online, 23 May 2017, “LEGO Porsche Crash Test in Slow-Motion”, [online] Available at: https://youtu.be/dCPWPj4JHqg

[12] Blender [Software], https://www.blender.org

[13] R. Luijkx. (16 Jun 2015). Using LoCo for Multi Run Simulations. Presented at the 10th European LS-DYNA Conference 2015. [Online]. Available: https://www.dynamore.de/de/download/papers/dynamore/de/download/papers/2015-ls-dyna-europ/documents/sessions-d-5-8-1/using-loco-for-multi-run-simulations

[14] LDraw OMR (Official Model Repository), http://omr.ldraw.org

[15] BrickLink Limited (2018). Studio (Version 2.0.1_93) [Software], https://www.bricklink.com/v3/studio/download.page