The Concept of a Physical Modelleling Language for Engineered Components "SMILE"

Prof. Dr.–Ing. Darius Friedemann, Dipl.-Ing. Jörg Rademann

15.01.2020

Hochschule für Technik und Wirtschaft Berlin
Fachbereich 2, Ingenieurwissenschaften - Technik und Leben
Wilhelminenhofstraße 75a
12459 Berlin
Joerg.Rademann@HTW-Berlin.de / +49 30 5019-3854
Darius.Friedemann@HTW-Berlin.de / +49 30 5019-3676

Introduction

In the development of motor vehicles and their components, the proportion of investigations carried out using computer numerical simulations has been increasing continuously for the last years. One reason for this is the shortening of development cycles, while the requirements for the components and vehicles are getting more complex. Furthermore, simulations in various development disciplines offer great cost advantages compared to the construction and testing of a prototype.

Due to the large number of different development disciplines of vehicle components (example of a forming simulation in Figure 1), it is common to model and test a component with different software tools. The use of different tools is partly due to differing interests regarding the result of a simulation (e.g. NVH versus stress analysis) or due to different strengths and weaknesses of the individual software tools. In the past decades, many different simulation programs have been developed simultaneously, each optimized for a specific application. It is expected in the near and medium term future that a wide variety of software solutions will still be used in the entire development process of a component. The fact that in the course of time, many programs have been developed into multidisciplinary software solutions that can be used for different applications is probably not going to change this fundamentally. All programs have one thing in common. The user needs a comprehensive understanding of the specific software, general know how of numerical simulations and special knowledge regarding the field of application and the simulation software used. When switching to another program within the same development discipline, the user has to build up extensive, software-specific knowledge, which is why creating a new model will initially involve a high amount of effort. The expertise regarding an application discipline and numerical simulations in general is no longer sufficient for model creation. To carry out a successful development task, an engineer using simulations should ideally be an expert in three fields at the

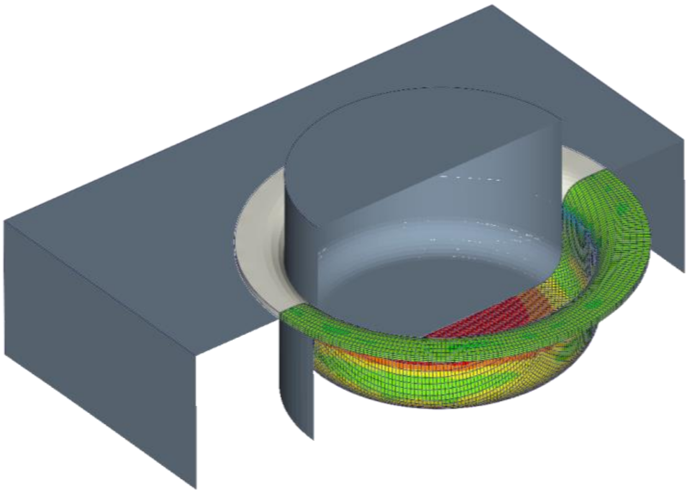

Figure 1: Example of a forming simulation

same time: *software understanding*, *numerical simulation* and subject-specific *development know-how.* If the necessary knowledge of the individual sub-areas could be separated (-> "democratization of simulation"), a lot of benefits could be drawn. First, the engineer could focus on the physical problem, regardless of numerical issues. Furthermore, this concept offers the possibility to save development time. Additionally it would be possible for engineering companies to employ specialists who are particularly well qualified in their respective fields. Therefore, the quality of development tasks can be increased.

This concept requires a standardized software-independent modelling language that describes a component solely from a physically point of view as the following will show. An example of an approach to resolve the described challenge is the meta language „Unified **S**imulation **M**odell**I**ng **L**anguag**E**"– **SMILE**. This language provides a syntactical basis for the description of a component in terms of its physical properties solely. The specific knowledge regarding a special simulation software and numerical simulation in general is not yet required. As soon as a simulation model has been built up with SMILE, it is possible to translate it automatically into the modeling language of any solver and start the corresponding simulation. The information about modeling standards and/or settings required by the solver itself can be defined in SMILE *modeling guidelines*. The translation of material cards for different solvers and fields of application is not possible for general reasons and usually requires individual case validation. Therefore, SMILE modelling deliberately assumes an existing material database for the solvers used. Translators that run script-based or integrated in preprocessors carry out the automatic conversion of a SMILE model into the format of a solver. These need to be provided for each software once. For the solvers ABAQUS and LS-DYNA, translators have already been created within the activities of a research project in order to perform a first proof of concept.

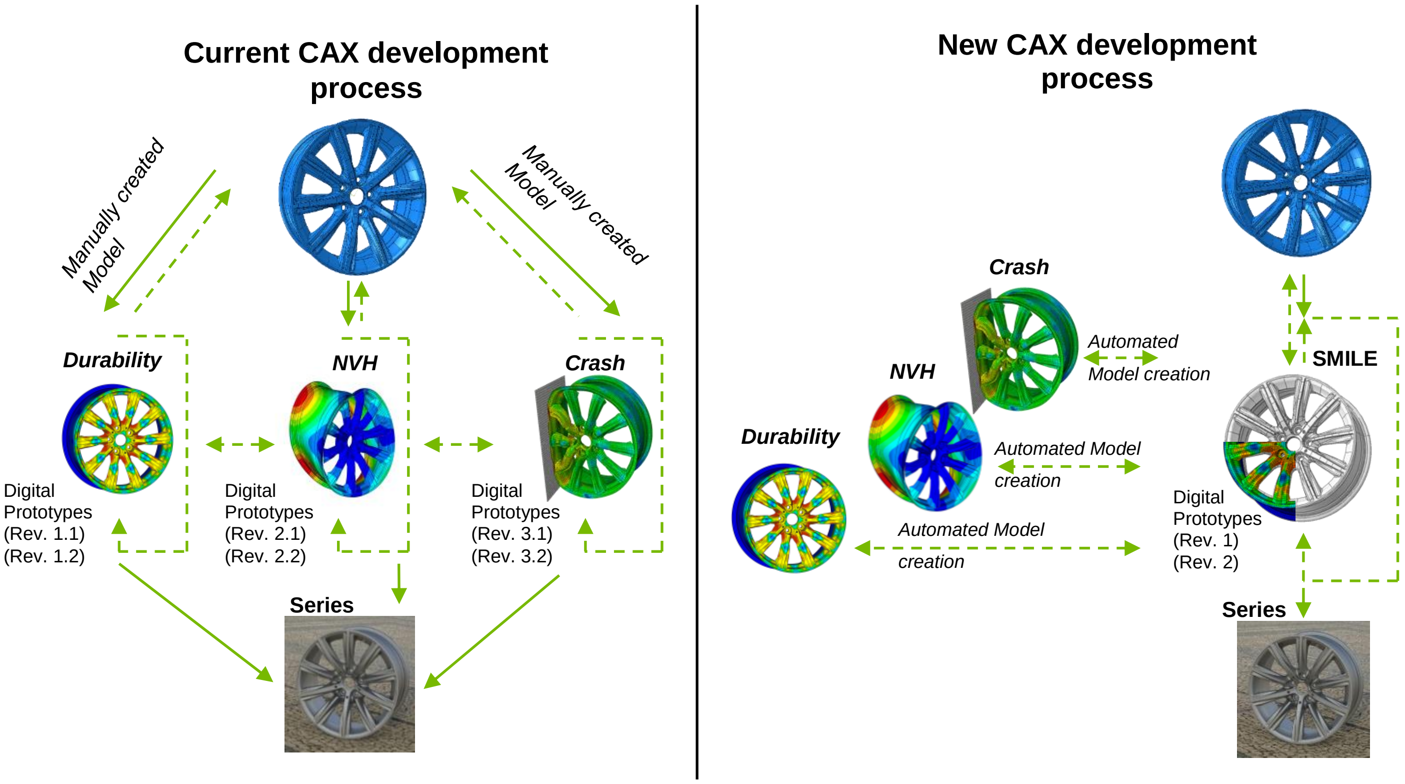

Figure 2: Comparison of the development processes without and with SMILE

Starting out from a digital prototype, an independent simulation model is created for each development discipline in the current process. In the individual disciplines, the model is optimized with regard to the different requirements.

The different optimization results lead to a compromise solution and thus to a revision of the digital prototype. The SMILE based development process is quiete more effective: In contrast to the current process, all necessary information from the different development disciplines is integrated into a SMILE model in addition to the geometric information of the digital prototype. From this standardized model, the various simulation models are automatically created and optimized. New design proposals are added as revisions to the SMILE model and the new simulation models are automatically created. This process is repeated until all requirements of the prototype are fulfilled. Figure 2 compares the different tasks of development processes with and without SMILE. In relation to the development process described, Figure 3 shows the potential for reduction of the workload by using SMILE. The additional effort when only a small number of development disciplines are used is caused by a slightly increased modeling effort of *model file* and *configuration file*[1]. In most

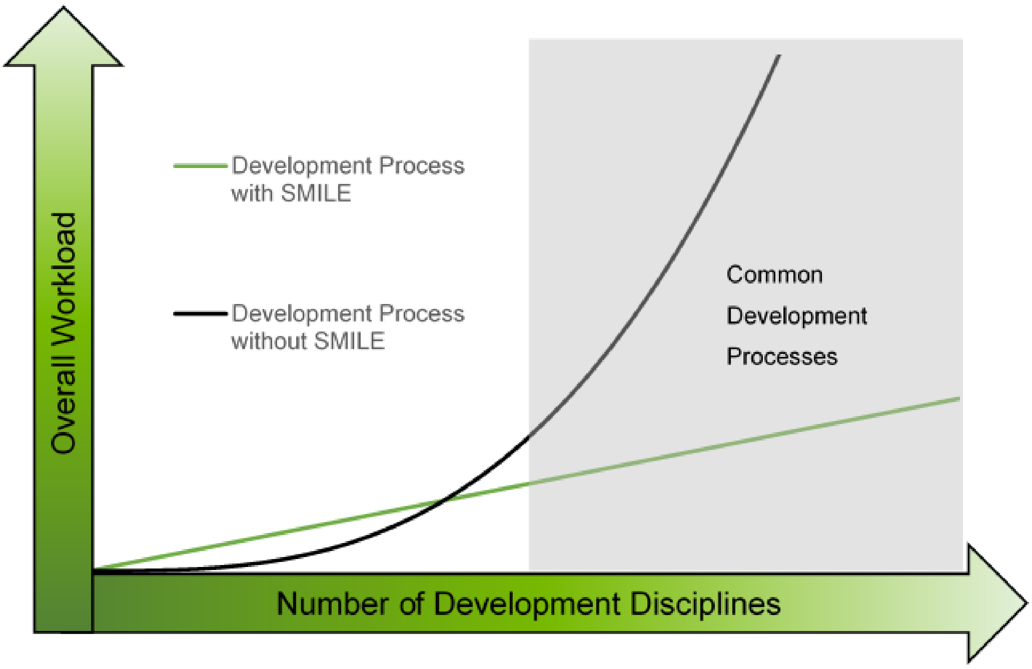

Figure 3: Comparison of the Workload in Development Processes

development processes, the additional modeling effort is quickly compensated by the automated model creation and thus the overall workload can be reduced. Figure 4 shows a detailed project plan, which is intended to give a rough summary of the SMILE project. Currently more complex examples of

[1] The smile *model file* and *configugation file* are explained in more detail in the following chapter.

automotive components are being created to further optimize the syntax of SMILE as well as the automated translators for ABAQUS and LS-DYNA. In the last phase of the project, a full vehicle car body will be modelled with SMILE, in order to prove the capability of SMILE for commercial usage in the automotive industry.

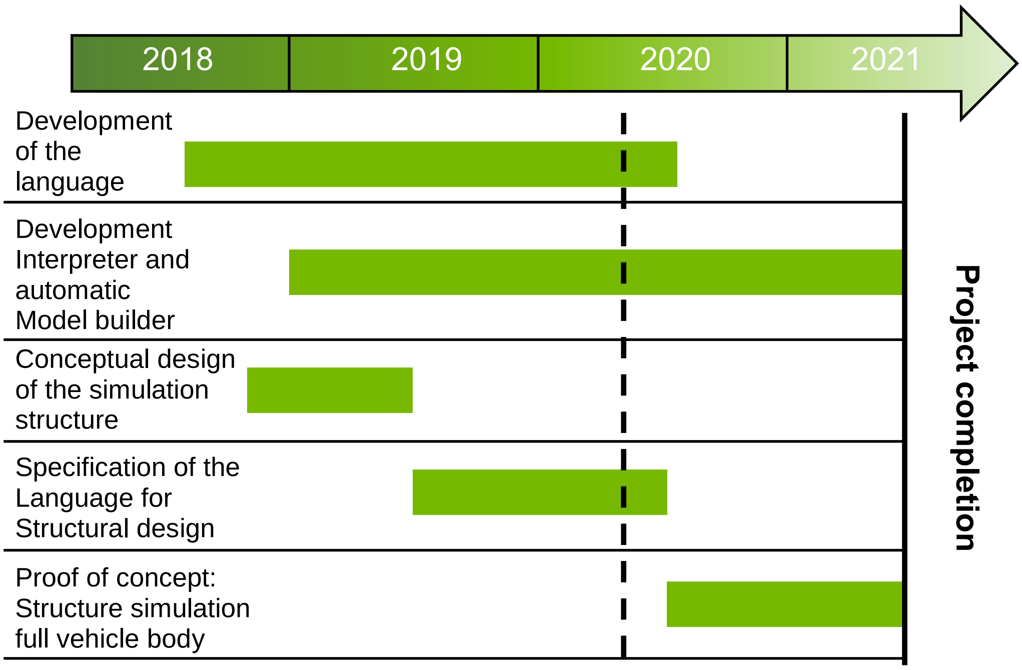

Figure 4: Project Timetable

Structure and Process

The syntax of SMILE is based on XML, as this is a completely standardized language that can be used independently of the operating system. Additionally, the syntax of an XML file can be handled with most text editors. Further advantages of XML are the continuous development of the language, the high acceptance and the fact that the language is machine readable as well as human readable. SMILE offers a language-like approach to describe simulation models. The number of the numerical simulation tools available is great. Therefore, SMILE has been developed with an open and easily extensible standard.

SMILE allows three different types of files to be created. A *model file*, a *configuration file* and *modeling guideline files*. The SMILE *model file* consists of an assembly and optional subassemblies which may consist of one or more components. All physical properties of the assembly can be defined independently of the exact shape of a geometry, because several FE-mesh or CAD geometries can be referenced in a SMILE *model file*. If different *model files* are required in a load case, they can be merged in the SMILE *configuration file*. In this file the load case itself is also defined, for example by initial and boundary conditions. In order to make sure, the application engineer does not

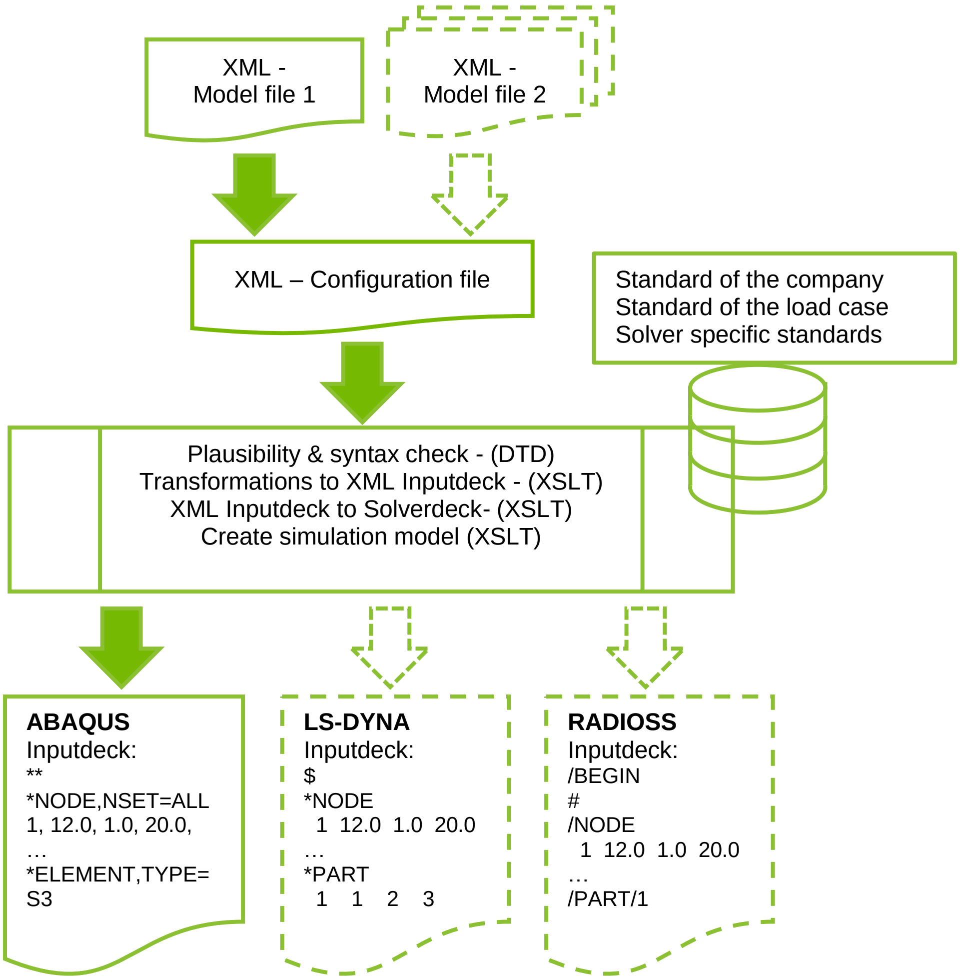

Figure 5: Complete translation process of a SMILE data set to a solver specific input deck

have to worry about solver-specific settings or idealizations, these are predefined as templates in the SMILE *modeling guidelines*. In the development process, different idealizations may be required depending on the application case. Therefore, it is possible to use different SMILE modeling guidelines for different load cases or application areas. These can be defined in different hierarchical levels to take varying specifications of different parties into account.

The first level could be the specifications of the solver manufacturer, followed and therefore potentially overwritten in parts by the specifications of the company and finally the specifications of the development team. The SMILE modelling guidelines provide a basis to take the different modeling approaches of the different application areas into account in automated translation. Since no solver-specific modelling technics of FEM are used in a SMILE model file, but idealizations still have to be made, the guidelines recommended by the software manufacturer can be used by default. In these guidelines, for example, element type, element properties and contact properties are pre-defined taking into account the use case. However, each user is free to use different default settings for his own load case. SMILE can also be used to create user specific modeling guidelines. Figure 5 shows how the process for an automated translation into simulation models can look like with the different SMILE files. The process flow diagram shows the different file types (model file, configuration file and modeling guideline) and how they interact in the complete process. It is also easy to see that by separating the physical, geometrical and numerical properties, the different tasks can be separated in terms of personell.