Template-driven management of model and loadcase variants for LS-DYNA simulations

Irene Makropoulou[1]

[1]BETA CAE Systems

1 Introduction

In recent years, crash and safety simulations have reached a very high level of accuracy in the prediction of the crashworthiness of the vehicle and the probability of injury for occupants and pedestrians under a multitude of loading scenarios. Among several factors, this achievement is also attributed to the fine resolution of finite element models that enables the precise representation of even the smallest parts and geometric features affecting the simulation results and the increase in the number of simulated loadcases. However, accuracy does come with a cost: Model size and variability have considerably increased, together with the number of loading scenarios that need to be simulated on each model variant.

From the definition of the different model variants from the CAD structure and the subdivision of the simulation models into functional sub-assemblies, to the set-up of the numerous different types and flavors of loadcases, there’s one thing in common: The modular management of the model at each phase of its lifecycle. At early phases, modules are as small as CAD parts. Later, modules become functional sub-assemblies which, for crash and safety loadcases, are handled as include files. Although the modular way of work is the only reliable method to enable parallel work on different areas of the model, it increases the “administration cost” during the simulation preparation, as the pool of different modules (be they parts or include files) needs to be consumed by higher level structures in a way that facilitates data reuse and enables traceability throughout the complete digital thread of a simulation model.

BETA CAE Systems’ Suite of applications addresses these challenges during model build and loadcase set-up with its Modular Model and Run Management solutions, that facilitate the handling of the different model and loadcase variants in a way that maximizes data reuse and enables traceability from part to simulation run, while mitigating the “administration cost” with the extensive use of templates. From the population of CAE subsystems based on the CAD structure, to the definition of different subsystem variants, vehicle configurations and loadcases, templates act as recipes that hold the instructions on which ingredients to use and how, in order to complete each given task.

This work discusses the definition and use of templates during model build and loadcase set-up, focusing on three key phases of the simulation preparation: First, the CAD to CAE structure mapping during Subsystem definition from PDM/PLM structures. Second, the handling of Subsystem variants in the scope of the different vehicle configurations. Third, the handling of parametric include files for the definition of Loadcases. Insights are given on the methods used for the adaptation of the templates to the specifications of each model in hand and how these are finally interpreted into LS-DYNA keywords behind the scenes, by making use of parameters, transformations and ID management techniques.

2 Typical Management of Crash Simulations

Crash simulations, due to the size and complexity of the models and the structure of CAE teams, are traditionally managed using include files. To promote efficient data sharing within the team, a pool of include files is typically created and maintained, storing all model, loadcase, and material includes. Using this pool, simulation master files are either compiled manually or automatically through in-house scripts or commercial tools.

However, this approach assumes that the include file is the smallest model unit tracked during simulation management. This simplification hampers overall model traceability. Each include file of the model represents a variant of a module, consisting of specific parts as outlined in the Digital Mock-Up exported from the PDM/PLM system. Losing the connection between parts and include files can complicate the updating process when new CAD data becomes available, delay communication between simulation engineers and the design team, and obstruct error impact analysis when needed.

At system level, a model variant (i.e. vehicle configuration) is composed of a list of specific include file variants. Managing which include file variants to use in each model variant becomes a challenging task, particularly for teams handling products with numerous variation options.

At the simulation level, different loading scenarios introduce additional Loadcase Variants, each comprising its own set of include files.

The shortcomings of traditional include file management in crash simulations, such as these and others, are effectively addressed in BETA's Modular Run Management environment through the use of Model and Loadcase Configuration Templates.

3 Templates definition

The term "template" generally refers to a predefined structure or form used as a guide for creating something new. A key feature of templates is their ability to be built once and reused multiple times. In the Modular Run Management environment, templates play a crucial role in defining and setting up simulation model and loadcase variants. These templates are typically created at the start of a project and serve as the foundation for generating LS-DYNA model includes and compiling the LS-DYNA master file. There are two distinct types of templates:

- Model Configuration Tables and
- Loadcase Configuration Tables

3.1 Model Configuration Tables

Managing model variability poses a significant challenge for engineering teams. While different configurations of the same model often have several differences, they also share many common elements. Reusing the common data across various model configurations can result in substantial time savings during the preparation of simulation models and is crucial for maintaining comprehensive data traceability throughout the process.

Figure 1 shows an excerpt of a vehicle configurations table.

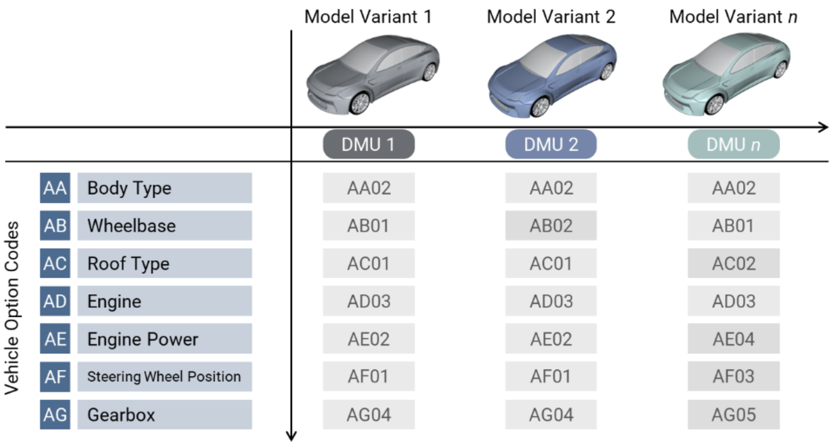

Fig.1: Vehicle configurations table

This table outlines the configuration of three different model variants based on a set of Vehicle Option Codes which dictate how a vehicle is built. Although there are usually more than 100 codes associated with a model variant, only a small subset is relevant to the CAE models. Furthermore, even among these few, not all impact every include of the model. For instance, the "Roof Type" influences the body-in-white, but has no effect on the trunk.

These codes, together with details on which includes they impact, are the first block of information in the Model Configuration Table.

At the same time, each of the model variants has its own Digital Mock-Up (DMU) in the PDM/PLM system. The product structure of the complete vehicle, as it is downloaded from the PDM/PLM system, is organized into groups of parts that facilitate the needs of product design and manufacturing, but not necessarily the needs of CAE teams. An example is shown in Figure 2.

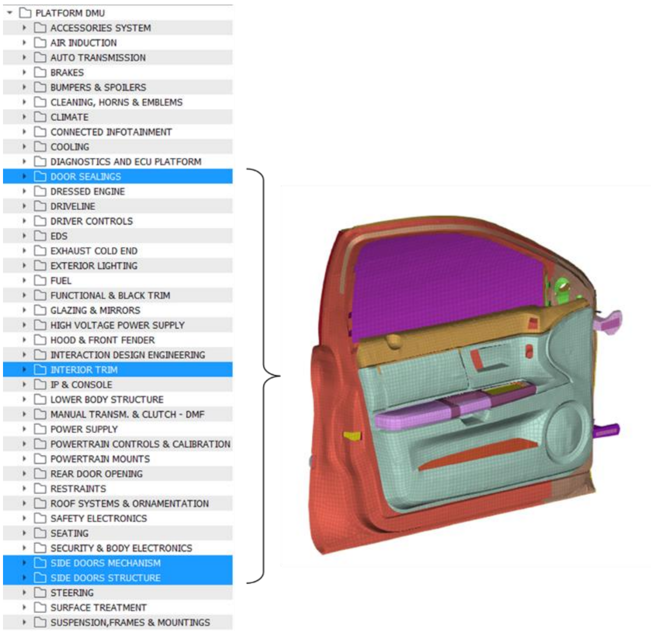

Fig.2: CAD to CAE structure for the front left door include

The list on the left shows the first level structures of the DMU of a model variant. The highlighted branches are the ones that relate to the front left door include file. This association of DMU branches with the CAE includes is the second block of information in the Model Configuration Table.

Finally, some includes may need to be transformed in order to be included in the model variant CAE model. This transformation may relate to their position, their IDs, or their units. This association of CAE includes with **`*DEFINE_TRANSFORMATION`** ids, ID offsets and unit transformation factors is the last block of information in the Model Configuration Table.

Figure 3 shows a schematic representation of a Model Configuration Table, which is defined in the Modular Environment as an Excel spreadsheet.

Model Variant Name	Link to DMU Structure	Vehicle Option Codes	*INCLUDE_TRANSFORM Info
LS-DYNA Includes	Groups of DMU Structure	Dependency to Vehicle Option Codes	TRAN ID, ID Offsets and Unit Transformation Info

Fig.3: Schematic representation of a Model Configuration Table

This table is automatically converted to an ANSA file that describes the model variant and its contained include files, which, in the Modular Environment are referred to with the terms “Simulation Model” and “Subsystems” respectively (Fig. 4)

	Variant Info	DMU Structure	Transformation	ID Offsets
Simulation Model	Model Variant 2			
100_biw	AA02_AB02_AC01_AF01			
220_door_fl	AA02			
230_door_fr	AA02			
240_door_rl	AA02			
250_door_rr	AA02			
260_hood	AA02			
270_trunk	-			
305_ip	AA02			
321_seat_fl	AF01		9,000,001	9,000,000
...	-			

Fig.4: Model Configuration Table in ANSA

3.2 Loadcase Configuration Tables

The management of the different loadcases is another challenge for simulation engineers. The loadcase setup in LS-DYNA simulations usually requires the collection of the relevant includes that represent solution controls, loadcase settings, contacts, initial conditions, etc. These includes may be standardized according to the loadcase type or may have certain dependencies on the model in hand. The include files used in the loadcases are usually parametric, i.e. define certain keyword fields based on parameters that are defined in the master file as `*PARAMETER` and `*PARAMETER_EXPRESSION` keywords.

Loadcase Configuration Tables compile all this information along with details related to spatial transformations, ID offsets, unit scaling and parameter values within an ANSA file. Figure 5 shows a schematic representation of a Loadcase Configuration Table.

Loadcase Name	Loadcase Variant Info	*INCLUDE_TRANSFORM Info	*PARAMETER Info
LS-DYNA Includes	Model and Loadcase-specific Variant	TRAN ID, ID Offsets and Unit Transformation Info	Parameter Values and Expressions

Fig.5: Schematic representation of a Model Configuration Table

Figure 6 shows a Loadcase Configuration Table in ANSA. Through this layout, simulation engineers can quickly review the complete list of loadcases, compare loadcases to identify unintentional differences and eventually use the loadcase definition as a basis for the creation of any simulation.

	Variant Info	Transformation	ID Offsets	Parameters
FMVSS_208				
000_controls				ENDTIM=150
004_initial_velocity				INVEL=13.33
012_barrier_transformation	rigid wall	6,000,001		
013_barrier	rigid wall			POSX=-360
021_dummy	H350			
EURO_NCAP_MPDB				
000_controls				ENDTIM=150
012_barrier_transformation	mpdb			POSX=-1012
013_barrier	mpdb	6,000,001	70,000,000	
...				

Fig.6: Loadcase Configuration Table in ANSA

4 Use of templates

The templates are an integral part of the Modular Environment solution offered by BETA. The template-driven management of simulations takes place in an environment where SPDRM is used for simulation data and process management, KOMVOS is used as a front-end for any data or process management activities and finally ANSA is the primary pre-processor (Fig.7)

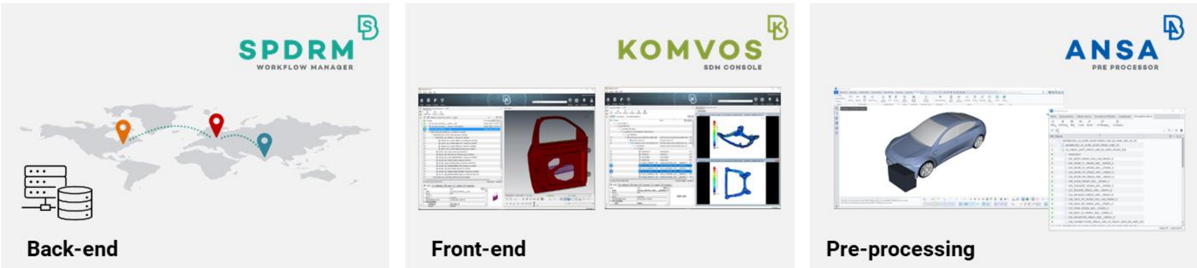

Fig.7: Key components of the Modular Environment

Figure 8 shows the complete workflow for the management of Crash Simulations within the Modular Environment.

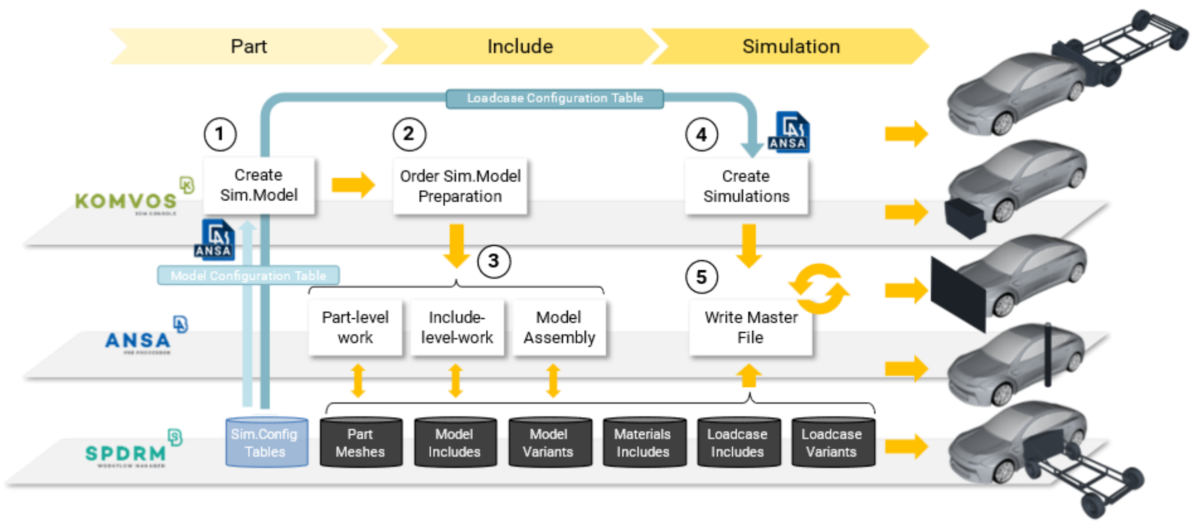

Fig.8: Management of Crash Simulations in the Modular Environment

The process starts in KOMVOS, where the modelling team engineer initiates the creation of the Simulation Model using the Model Configuration Table. At this stage, the product structure from the DMU is imported and divided into CAE Subsystems based on the Model Configuration Table.

The next step, also performed in KOMVOS, involves usually the same user initiating the preparation of the Simulation Model. This process is primarily executed in ANSA, starting with part-level work, progressing to the preparation of include files, and concluding with the integration of these includes into the model assembly.

Once all the simulation components are in place, simulation engineers proceed to create simulations based on the Loadcase Configuration Table. The master file composition is completed in ANSA, which generates the LS-DYNA keyword file.

	Variant Info	Transformation	ID Offsets	Parameters
Simulation Model	Model Variant 2			
100_biw	AA02_AB02_AC01_AF01			
220_door_fl	AA02			
230_door_fr	AA02			
240_door_rl	AA02			
250_door_rr	AA02			
260_hood	AA02			
270_trunk	-			
305_ip	AA02			
321_seat_fl	AF01	9,000,001	9,000,000	
EURO_NCAP_MPDB				
000_controls				ENDTIM=150
012_barrier_transformation	mpdb			POSX=-1012
013_barrier	mpdb	6,000,001	70,000,000	
...				

Fig.9: Crash Simulation Configuration in the Modular Environment

Naturally, this process is not a one-time task. The final step is repeated hundreds, if not thousands, of times throughout each project, during the simulation looping phase. ANSA significantly streamlines this activity by ensuring that model and loadcase configuration templates are not static “recipes” but rather dynamic links to SPDRM’s database. Through this live connection, simulation engineers can check the status of includes in the data repository, apply available updates, or load any of the includes and modify them to create new simulation iterations.

	Variant Info	Study Version
Simulation Model	Model Variant 2	0001
100_biw	AA02_AB02_AC01_AF01	001
220_door_fl	AA02	001
230_door_fr	AA02	001
240_door_rl	AA02	001
250_door_rr	AA02	001
260_hood	AA02	001
270_trunk	-	001
305_ip	AA02	001
321_seat_fl	AF01	001
EURO_NCAP_MPDB		01
000_controls		01
012_barrier_transformation	mpdb	01
013_barrier	mpdb	01
...		

DM Status Indication

- Does not exist in DM
- Exists in DM and is the latest available
- Exists in DM and has updates
- Was modified and will be saved

Fig.10: Live indication of status in DM for crash simulations and their contents

5 Summary

This work introduces a template-driven approach for managing model and loadcase variants in crash simulations, as implemented in the Modular Environment of BETA. This methodology offers several key advantages for engineering teams.

First, it facilitates data reuse across different model variants, significantly reducing CAE turnaround time by streamlining the preparation of include variants.

Second, it ensures comprehensive traceability throughout the digital thread of model and simulation data - from individual parts to simulation runs and results. This way, minor DMU updates can be easily incorporated in the include files. Furthermore, the traceability not only supports efficient root cause and error impact analysis but also boosts engineers' confidence in the accuracy of the simulations.

Although the focus of this study is on LS-DYNA simulations, the methodology is versatile enough to be applied across various disciplines, making it suitable for broader CAE team adoption.

Last, ANSA's advanced capabilities, such as the automated translation of templates into LS-DYNA master files and seamless updates based on the data repository, significantly enhance the efficiency of simulation engineers.