

2014

# A NEW FRAMEWORK FOR SUPPORTING AND MANAGING MULTI-DISCIPLINARY SYSTEM- SIMULATION IN A PLM ENVIRONMENT

Author: Dipl. Ing. (BA) Michael Mahler BSc (Hons)

A thesis submitted to the University of Gloucestershire in accordance with the requirements  
for the degree of PhD in the Faculty of Media, Arts and Technology

May 2014 of submission

## ABSTRACT

In order to keep products and systems attractive to consumers, developers have to do what they can to meet growing customers' requirements. These requirements could be direct demands of customers but could also be the consequence of other influences such as globalization, customer fragmentation, product portfolio, regulations and so on. In the manufacturing industry, most companies are able to meet these growing requirements with mechatronic and interdisciplinary designed and developed products, which demand the collaboration between different disciplines. For example, the generation of a virtual prototype and its simulation tools of a mechatronic and multi-disciplinary product or system could require the cooperation of multiple departments within a company or between business partners. In a simulation, a virtual prototype is used for testing a product or a system. This virtual prototype and test approach could be used from the early stages of the development process to the end of the product or system lifecycle. Over years, different approaches/systems to generating virtual prototypes and testing have been designed and developed. But these systems have not been properly integrated, although some efforts have been made with limited success. Therefore, the requirement exists to propose and develop new technologies, methods and methodologies for achieving this integration.

In addition, the use of simulation tools requires special expertise for the generation of simulation models, plus the formats of product prototypes and simulation data are different for each system. This adds to the requirements of a guideline or framework for implementing the integration of a multi- and inter- disciplinary product design, simulation software and data management during the entire product lifecycle.

The main functionality and metadata structures of the new framework have been identified and optimised. The multi-disciplinary simulation data and their collection processes, the existing PLM (product lifecycle management) software and their applications have been analysed. In addition, the inter-disciplinary collaboration between a variety of simulation software has been analysed and evaluated. The new framework integrates the identified and optimised functionality and metadata structures to support and manage multi- and inter-disciplinary simulation in a PLM system environment.

It is believed that this project has made 6 contributions to new knowledge generation: (1) the New Conceptual Framework to Enhance the Support and Management of Multi-Disciplinary System-Simulation, (2) the New System-Simulation Oriented and Process Oriented Data Handling Approach, (3) the Enhanced Traceability of System-Simulation to Sources and Represented Products and Functions, (4) the New System-Simulation Derivation Approach, (5) the New Approach for the Synchronisation of System Describing Structures and (6) the Enhanced System-Simulation Result Data Handling Approach.

In addition, the new framework would bring significant benefits to each industry it is applied to. They are: (1) the more effective re-use of individual simulation models in system-

simulation context, (2) the effective pre-defining and preparing of individual simulation models, (3) the easy and native reviewable system-simulation structures in relation to input-sources, such as products and / or functions, (4) the easy authoring-software independent update of system-simulation-structures, product-structures and function-structures, (5) the effective, distributed and cohesive post-process and interpretation of system-simulation-results, (6) the effective, easy and unique traceability of the data which means cost reductions in documentation and data security, and (7) the greater openness and flexibility in simulation software interactions with the data holding system.

Although the proposed and developed conceptual framework has not been implemented (that would require vast resources), it can be expected that the benefits in 7 above will lead to significant advances in the simulation of new product design and development over the whole lifecycle, offering enormous practical value to the manufacturing industry.

Due to time and resource constraints as well as the effort that would be involved in the implementation of the proposed new framework, it is clear there are some limitations to this PhD thesis. Five areas have been identified where further work is needed to improve the quality of this project: (1) an expanded industrial sector and product design and development processes, (2) parameter oriented system and production description in the new framework, (3) the improved user interface design of the new framework, (4) the automatic generation of simulation processes and (5) enhancement of the individual simulation models.

Keywords:

System-Simulation; multi-disciplinary Simulation; inter-disciplinary Simulation; co-simulation; System Engineering; Simulation Data Management; PLM; simulation process

## DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of the University of Gloucestershire and is original except where indicated by specific reference in the text. No part of the thesis has been submitted as part of any other academic award. The thesis has not been presented to any other education institution in the United Kingdom or overseas.

Any views expressed in the thesis are those of the author and in no way represent those of the University.

Signed: 

Date: 2<sup>nd</sup> March 2014



## ACKNOWLEDGMENT

The biggest and most important thanks have to be to my wife. Her role during the research project was both influential and significant. She provided endless encouragement for me to persevere and bring the research project to its conclusion. She helped in many ways such as driving me to the next meeting while I worked at my laptop, all the while feeling lonely tending to our daughter and home, having lost part of me to this project. Nevertheless, she took on all this additional work selflessly. Special thanks and best wishes go to my wife for her assistance. Finally, thank you to my wife and daughter; I am sorry for having spent the time which was dedicated to you, working on my research.

The research project had to achieve the complexity and respect of a significant number of influences relevant for the new framework. The handling of such complexity and influences was often a challenge. In order to bring the research work into a reasonable format, the support and help of supervisors and advisors was an important factor. Thanks to Dr. Shujun Zhang and Dr. Kevin Hapeshi for their support and assistance.

The research project was based on good and broad expertise as well as knowledge in simulation and product development processes acquired during my employment as consultant at Siemens Industry Software GmbH & Co.KG. The case examples are based on this expertise and knowledge. As such, the actual commencement of the research project was much earlier than the official start at the University of Gloucestershire. My employment at Siemens Industry Software GmbH & Co.KG has played a significant part in helping me bring the project to fruition. Most managers and colleagues understood the impacts of the research project and gave important assistance before and during the process. Therefore thanks to my managers and colleagues for their assistance.

## TABLE OF CONTENTS

Abstract .....	2
Declaration .....	4
Acknowledgment .....	5
Table of Contents .....	6
Abbreviations .....	11
List of Figures .....	13
List of Tables .....	16
1 Introduction .....	17
1.1 Introduction .....	17
1.2 Aim and Objective .....	19
1.3 Discussion of the Problems .....	19
1.4 The Discussion of the Idea of a new Framework for Supporting and Managing Multi-Disciplinary Simulations .....	22
1.5 Organisation of the Thesis .....	23
2 Literature Review .....	25
2.1 Introduction .....	25
2.2 Multi-Disciplinary Simulation .....	26
2.3 Simulation Data Management .....	28
2.4 Product Data Management .....	34
2.5 Product Lifecycle Management .....	38
2.6 Data Management of Multi-Disciplinary Simulations .....	41
2.7 The System Engineering Methodology .....	46
2.8 Problem Discussion .....	51
2.8.1 Functions for the Support of Multiple Simulation Authoring Tools .....	51
2.8.2 Functions for the Support of the Cooperation and Interaction between multiple simulations .....	52
2.8.3 Functions in the Support of Different System or Product Descriptions .....	53
2.8.4 Functions for the Support and Interaction Between Different Development Disciplines .....	54
2.9 Proposition of a New Framework .....	55

3	Research Methods .....	56
3.1	Informed Topics .....	56
3.1.1	Ontological Viewpoint .....	56
3.1.2	Ontological Base.....	57
3.1.3	Ontological Structuring of the Research Project .....	59
3.2	Used Methodology and Methods.....	63
3.2.1	Data Sampling for the Data Acquisition of Data Management Technologies.....	65
3.2.2	Data Sampling for the Analysis of the Case Examples and Case Study .....	66
3.2.3	Conceptualisation of the New Framework .....	67
4	Data Acquisition of the Simulation Supporting Systems and Technologies in TEAMCENTER .....	68
4.1	Data Acquisition of the Basic Data Model of TEAMCENTER.....	68
4.2	Data Acquisition of the Simulation Data Model in TEAMCENTER .....	71
4.3	Data Acquisition from the Revision of CAE Items .....	75
4.4	Data Acquisition of Datasets Technology in TEAMCENTER.....	77
4.5	Data Acquisition of the Standard Simulation Data Model of TEAMCENTER.....	77
4.6	Data Acquisition of the Relationships to Organise the Simulation Data Model .....	79
4.6.1	Relationships between the Simulation Data Model .....	79
4.6.2	Relationship between CAE ITEM Revision and CAE Datasets .....	83
4.6.3	Relationship between Different Simulations .....	84
4.7	Export of Files and Data for External Processes from the TEAMCENTER Database.....	85
4.8	Data Acquisition of the Behaviour Models Technology.....	89
4.8.1	Discussion: The Behaviour Models Technology in the Context of Mechatronic Development Processes.....	90
4.8.2	Discussion: The Behaviour Models Technology in Context to System Engineering.....	93
4.8.3	Data Acquisition of the Behaviour Models Technology in TEAMCENTER.....	95
5	Case Studies of Management of the Data of Multi-Disciplinary and Inter-Disciplinary Simulation .....	100
5.1	Simulation Process Basics.....	101

5.1.1	Pre-Process .....	102
5.1.2	Solve-Process .....	105
5.1.3	Post-Process .....	106
5.2	Case 1: Creation of a Moulding-Tool-Simulation-Vision for Moulding-Tool-Optimization .....	107
5.2.1	Discussion of Moulding Tool Simulation .....	107
5.2.2	Discussion of the Simulation Process and Information Flow in Case 1 110	
5.2.3	Discussion about the Used and Generated Files, Data and Information of Case 1.....	115
5.2.4	Data Management of Case 1 with TEAMCENTER.....	120
5.3	Case 2: Improved Development Process of Extrusion Section Production Tools by Simulation and Design Model Interaction .....	125
5.3.1	Discussion of Extrusion Section Production Process .....	126
5.3.2	Discussion of the Simulation Process and Information Flow in Case 2 128	
5.3.3	Discussion about the Used and Generated Files, Data and Information of Case 2.....	130
5.3.4	Data Management of Case 2 with TEAMCENTER.....	133
5.4	Case 3: System-Simulation of a Lift Gate.....	139
5.4.1	Discussion about The Lift Gate System .....	139
5.4.2	Discussion of the System and Subsystem-Simulation of Case 3.....	143
5.4.3	Discussion of the Simulation Architecture and System-Simulation Generation of Case 3.....	146
5.4.4	Data Management of Case 3 with TEAMCENTER.....	150
5.4.5	Empirical Validation of the New Framework and Case 3 .....	161
5.5	Case 4: Project: "Interdisciplinary Model-Based Development Process" .....	164
5.5.1	Discussion of the Electrical Front Door Windows Lifter System.....	164
5.5.2	Discussion of the System- and Subsystem-Simulation of Case 4.....	166
5.5.3	Discussion of the Simulation Architecture and System-Simulation Generation of Case 4.....	174
5.5.4	Data Management of Case 4 .....	184
5.5.5	Empirical Validation of the New Framework and Case 4 .....	186
6	A New Framework For Improving the Support and Management of Multi-Disciplinary Simulation Data in a PLM Environment .....	192

6.1	Institution of a System-Simulation Structure .....	192
6.1.1	Analysis of the Cases .....	192
6.1.2	The System-Simulation Structure Approach .....	195
6.1.3	Technical Detailing of the Approach .....	200
6.2	Combination of System-Simulation Structure with Process-Oriented Simulation Processes .....	205
6.2.1	Analysis of the Cases .....	206
6.2.2	The Process-Oriented Simulation Approach.....	208
6.2.3	Technical Detailing of the Approach .....	211
6.3	Collaboration of the System-Describing Structures .....	214
6.3.1	Analysis of the Cases .....	214
6.3.2	The Structure Relationship Approach .....	215
6.3.3	Technical Details of the Approach.....	222
6.4	Institution of an System-Simulation Result Structure.....	232
6.4.1	Analysis of the Cases .....	232
6.4.2	The System-Simulation Result Structure Approach .....	233
6.4.3	Technical Detailing of the Approach .....	236
6.5	The Architectural Concept of the New Framework.....	238
6.5.1	Analysis of the Cases .....	239
6.5.2	The Architectural Concept Approach.....	240
6.6	Evaluation of the New Framework.....	248
6.6.1	Benefit of Traceability .....	248
6.6.2	Benefit of Openness for Multiple Simulation Tools and Simulation Middleware Tools.....	250
6.6.3	Benefit of Supporting the Individual Experts of Individual Simulation Model Generation and System-Simulation Model Generation.....	251
6.6.4	Benefit of Supporting Individual Experts of Individual Simulation Model Post-Processing and System-Simulation model Post-Processing .....	252
6.6.5	Benefit of Synchronization with Product and Function Structure.....	253
6.6.6	Benefit of Derivation of System-Simulation Structures based on Function or Product Structures.....	255
6.6.7	Benefit of System-Simulation Structure as 150% Structure .....	257

6.7	Critiques on the New Framework Approach.....	258
6.7.1	Missing Interface Descriptions .....	259
6.7.2	Missing Object-Oriented Visualisations of the System-Simulation .....	260
6.7.3	Missing Framework for Process-Oriented Simulation Processes .....	260
7	Conclusion and Further Work .....	262
7.1	The Research Objectives.....	262
7.2	Contributions to the New Knowledge Generation .....	263
7.3	Contribution to the Practice .....	267
7.4	Further Research .....	268
8	References.....	271
9	Appendix.....	284
9.1	Appendix A Systematic Literature Review.....	284
9.1.1	Selection of Literature .....	285
9.1.2	Quality and Relevance of the Literature .....	286
9.2	Appendix B Work Plan.....	293
9.3	Appendix C Enhanced Graphics of the PHD thesis.....	295
9.4	Appendix D Published Paper .....	323
9.5	Appendix E Patent Confirmation.....	343

## ABBREVIATIONS

API	application programming interface
BOM	bill of material
CAD	Computer aided design
CAE	Computer aided engineering
CAM	Computer aided manufacturing
CAPP	Computer aided process planning
CAx	Computer aided x
CFD	computational fluid dynamics
DMU	digital mock up
EBS	equation based simulation
EDM	engineering data management
ERP	enterprice resource planning
FEA	finite element analysis
FEM	finite element method
FMI	functional mockup interface
FMU	functional mockup unit
IT	information technology
MBS	multi body simulation
PDM	product data management
PLM	product lifecycle management
SCM	supply chain management
SDM	simulation data management

SDPM	simulation data and process management
SLM	simulation lifecycle management
SPM	simulation process management



## LIST OF FIGURES

Figure 2-1 research topic and subtopics .....	25
Figure 2-2 SDPM Integration in a Complex System and Process Landscape (Source: Virtual Vehicle).....	37
Figure 2-3 Levels of System Engineering .....	47
Figure 2-4 Mini V-Model for Verification .....	48
Figure 2-5 Big V-Model .....	49
Figure 4-1 Basic Minimum Structure .....	69
Figure 4-2 Item revision – Dataset Structure .....	70
Figure 4-3 CAE Item Graphic (Kondragunta, 2010).....	74
Figure 4-4 Manual Data Structure Example (Kondragunta, 2010).....	75
Figure 4-5 CAE Item Data Structure .....	76
Figure 4-6 Standard Simulation Data Model.....	78
Figure 4-7 Simulation Data Structure Example .....	83
Figure 4-8 CAE Specification Relationship .....	84
Figure 4-9 CAE Include Relationship Example.....	85
Figure 4-10 Data Export and Import.....	89
Figure 4-11 Mechatronics System (Siemens Product Lifecycle Management Software Inc., 2012) .....	91
Figure 4-12 Model-Based Mechatronic Development Methodology (Mahler, Framework for System-Simulation based System Engineering Development Methodology, 2012) .....	93
Figure 4-13 Ordering of the Behaviour Data Model .....	96
Figure 4-14 Structure of Model-Based System Engineering Models .....	97
Figure 4-15 Matlab/Simulink Behaviour Model Interface Architecture (Siemens Product Lifecycle Management Software Inc., 2012).....	98
Figure 4-16 Behaviour Model Item Structure .....	99
Figure 5-1 Basic Simulation Process .....	102
Figure 5-2 Basic Pre-Process Workflow .....	103
Figure 5-3 Basic Pre-Process File System of NX Advanced FEM .....	104
Figure 5-4 Files in the Solve-Process .....	105
Figure 5-5 Post-Process Visualisation .....	106
Figure 5-6 Case 1: Moulding Process .....	108
Figure 5-7 Result of Phase I Case 1 .....	112
Figure 5-8 Result of Phase 2 Case 1.....	113
Figure 5-9 Result of Phase 3 Case 1.....	115
Figure 5-10 Phase 1 Simulation Structure of Case 1.....	116
Figure 5-11 Phase 2 Simulation Structure of Case 1.....	117
Figure 5-12 Simulation Structure of Case 1.....	119
Figure 5-13 Metadata Structure of Case 1 Phase 1.....	121
Figure 5-14 Metadata Structure of Case 1 Phase 1 and 2.....	123
Figure 5-15 Metadata Structure of Case 1 Phase 1, 2 and 3 .....	125

Figure 5-16 Extrusion Process (Udomphol & Technology, 2012) .....	127
Figure 5-17 Extrusion Process (WIKIPEDIA, 2012) .....	127
Figure 5-18 Simulation Process of Case 2 .....	130
Figure 5-19 File Structure of Case 2 .....	133
Figure 5-20 Metadata Structure of Case 2 Phase 1.....	136
Figure 5-21 Metadata Structure of Case 2.....	138
Figure 5-22 Car Lift Gate (Wuttke, Bohn, & Suyam-Welakwe, 2011).....	139
Figure 5-23 Lift Gate System Architecture .....	140
Figure 5-24 Supplier Integration (Link, 2012) .....	142
Figure 5-25 Simulation Discipline Architecture Case 3.....	144
Figure 5-26 Case 3: Simulation Tool Architecture Source .....	145
Figure 5-27 Simulation Model Architecture of Case Example 3 .....	150
Figure 5-28 Rigid Body Architecture of Case 3 .....	151
Figure 5-29 Metadata Architecture of the Flexible Body in Case 3 .....	153
Figure 5-30 Empirically Logical Matlab/Simulink Ordering.....	155
Figure 5-31 Behaviour Model Structure of Case 3.....	156
Figure 5-32 Integration of the Flexible Body.....	158
Figure 5-33 Temporary Interface Integration Model for Matlab/Simulink.....	159
Figure 5-34 Summary Data Management of Case 3 .....	160
Figure 5-35 Empirical Structure of Case 3 in the New Framework .....	163
Figure 5-36 Electrical Front Door Window Lifter System.....	165
Figure 5-38 Case 4: Simulation Discipline Architecture (Mahler, 2012) .....	167
figure 5-37 .....	167
Figure 5-39 Case 4: Simulation Tool Architecture (Mahler, 2012) .....	168
Figure 5-40 Assembling Version 1 of Case 4.....	170
Figure 5-41 Assembling Version 2 of Case 4.....	171
Figure 5-42 Assembling Version 3 of Case 4.....	172
Figure 5-43 Theoretical Modelica Simulation Model Structure of Case 4 .....	176
Figure 5-44 Example of a Cable Pull (dajo Autoteile, 2013).....	177
Figure 5-45 Seals in Case 4.....	179
Figure 5-46 Flexible Body Simulation Model of Case 4 .....	179
Figure 5-47 Flexible Body Simulation Process of Case 4.....	181
Figure 5-48 MODELISAR FMI for Model Exchange (Blochwitz & Otter, 2011).....	182
Figure 5-49 MODELISAR FMI for Co-Simulation (Blochwitz & Otter, 2011) .....	183
Figure 5-50 FMU-Structure of Simulation of Case 4 .....	184
Figure 5-51 Functions and Behaviour Models during the Development Process .....	187
Figure 5-52 Product and Simulation Data during the Development Process .....	188
Figure 5-53 Simulation Model Traceability of Represented and Sourced System.....	189
Figure 5-54 System-Simulation and Result Management .....	191
Figure 6-1 System-Oriented Simulation Process .....	194
Figure 6-2 Multi-Disciplinary Simulation Data Management Cores .....	196
Figure 6-3 Improved Approach of System-Simulation Source Dependencies.....	197
Figure 6-4 Existing Approach of System-Simulation Source Dependencies.....	199

Figure 6-5 Traceability-Example of the New Approach .....	200
Figure 6-6 Ordering of the System-Simulation Structure.....	203
Figure 6-7 Serial Simulation Process of Case 1 .....	206
Figure 6-8 Process-oriented Simulation Process.....	208
Figure 6-9 Existing Approach of Simulation Process Authoring Tools and Simulation Integrators.....	210
Figure 6-10 Improved Approach of Simulation Process Authoring Tools and Simulation Integrator.....	211
Figure 6-11 left a) New Approach – right b) Old Approach to Link System-Describing Structures .....	216
Figure 6-12 Source and Target Relationships .....	217
Figure 6-13 Target and Source Unique Relation of the Simulation Model for Usability in System-Simulation .....	218
Figure 6-14 System-Simulation Structure Derivation of Product or Functional Structure.....	219
Figure 6-15 Comparison of System-Describing Structures .....	221
Figure 6-16 Simulation Sources and Function Sources Dependencies.....	224
Figure 6-17 Sys-Sim Source and Target Relationship to Functions .....	226
Figure 6-18 Simulation Sources and Product Sources Dependencies.....	228
Figure 6-19 Sys-Sim Source and Target Relationship to Functions .....	229
Figure 6-20 New Approach of System-Simulation Result Structure Linking .....	233
Figure 6-21 Example of Simulation Result Traceability.....	234
Figure 6-22 Example of System-Simulation Result Management at Multiple System-Simulation Solve Runs.....	235
Figure 6-23 Old Approach of System-Simulation Result Structure Linking .....	236
Figure 6-24 Ordering Rules for Sys-Sim ITEM Revision and Sys-Sim Result ITEM Revision	238
Figure 6-25 Multi-Disciplinary Simulation Data Management Cores.....	241
Figure 6-26 Overview of the Improved System-Simulation Architecture .....	243
Figure 6-27 Architectural Concept of Commercial System-Simulation Data Management Applications.....	244
Figure 6-28 Improved Four Cores Technology Architectural Concept Approach.....	245
Figure 6-29 Improved Architectural Concept of Multi-Disciplinary Simulation Data Management.....	246
Figure 6-30 Management Architecture Achieved by the Ordering Rules.....	249
Figure 9-1 Relevance of the Literature.....	287

## LIST OF TABLES

Table 4-1 CAE Item Classes (Siemens Product Lifecycle Management Software Inc, 2012)....	73
Table 4-2 CAE Dataset Type (Siemens Product Lifecycle Management Software Inc, 2012)...	77
Table 4-3 CAE Relation Type Source: (Siemens Product Lifecycle Management Software Inc, 2012) .....	80
Table 5-1 Main NX Nastran Files.....	106
Table 5-2 Case 3: Table of Simulation Tools .....	145
Table 5-3 Independent Rigid Body Simulation Models of Case 3.....	147
Table 5-4 Case 4: Table of Simulation Tools .....	168
Table 6-1 Item Types used in the System-Simulation Structure .....	203
Table 9-1 Search Functions.....	284

# 1 INTRODUCTION

## 1.1 INTRODUCTION

This thesis will propose and design a new framework for supporting and managing multi-disciplinary system simulations in a PLM (Product Lifecycle Management) environment by exploring the complex relations between simulation sources, simulation model sources and system describing sources, as well system simulations.

Mr. Anton Huber, CEO of Siemens Industry Automation Division, said: “The integration of product development, simulation and validation is now at the top of our agenda.” (Siemens AG, 2011). In his speech at the Siemens global sales team conference, Mr. Chuck Grindstaff, CEO of Siemens PLM Software, said: “We are investing heavily in CAE (Computer aided engineering) [...] for multi-disciplinary analysis throughout the product definition cycle.” Many other senior managers also have similar thoughts. They all believe that it is necessary and important to design and develop complex products in a multi- and inter- disciplinary modelling and simulation environment. For example, Engelson (Engelson, 2000) said that the isolated subsystem simulation has become a thing of the past; the current trend is to simulate the increasingly complex physical systems and products as a composition of subsystems from multiple domains.

In the whole product lifecycle, these multiple disciplines normally include mechanics, electronics, hydraulics, pneumatics, controls, mathematics, chemistry and biology, etc. (Bharadwaj, 1998). In order to build a multi-disciplinary modelling and simulation environment, it is required to adopt an approach which can be used to develop a single disciplinary based simulation model into a comprehensive multi-disciplinary based simulation model (Ai, Chen, Wan, & Xiong, 2011). So, “several paths are being followed to enable communication between models of components from different domains” (van Beek, Rooda, Engell, & Zaytoon, 2000). Several previous projects have been carried out to deploy a single simulation environment based on a C-like language (Diamond, 1993). Such a simulation environment is expected to meet all simulation requirements of the different disciplinary simulation scenarios. This expectation will be seen critically by the users because of missing functionalities compared to disciplinary oriented simulation environments. However, each business’s functional department uses its own specialised simulation software tools in an isolated way. The various simulation software models have not been integrated yet (Zaeh & Baudisch, 2003). Hence, communication between different functional simulation models becomes a serious issue. In order to solve such a problem, it is necessary to design and develop vendor-independent APIs (application programming interfaces) to achieve the software system integration of various simulation models. The APIs can be used to solve communication problems between simulation models (MODELISAR consortium, January 26, 2010). These kinds of APIs will provide the interaction between different and multiple

simulation models. The simulation models could then be generated more independently by applicable departments using the most appropriate simulation authoring tools.

However, the use of different and multiple simulation models will lead to large amounts of simulation data. The simulation models require vast quantities of input data such as CAD (computer aided design) models or parameters. In a PDM/PLM (product data management/product lifecycle management) environment, the management of most product data will begin at an early stage of the product lifecycle. The PDM/PLM environment could provide important input to simulation model generation. Some PDM/PLM environments could also manage simulation data. These simulation models are then recorded together with different simulation model metadata (Brendel & Kühner, 2004).

At an early stage of a product lifecycle, mechatronic concepts of the product will be evaluated and analysed. The mechatronic product concepts will be used to optimise products. However in system engineering, the mechatronic product concept could also be used for the idealization of the product development process (Mahler, 2012). The mechatronic product concept could be an abstract, virtual prototype of the new product. At later stages of a product lifecycle, it is frequently required for the product design and development team to build a virtual prototype of a new product to reduce the product lead time (Mahler, 2012). In order to achieve this, the team has to identify and optimise the required simulation models using the simulation model metadata defined at the early stage of the product lifecycle. Then, the processes for simulation model data collection in both multi-disciplinary and inter-disciplinary environments need to be determined. However, there is no existing software available to help a team in a PLM environment achieve this goal (Ai, Chen, Wan, & Xiong, 2011; Vyatkin V. , Hanisch, Cheng, & Chia-Han, 2009).

So, it is not only theoretically meaningful, but also practically important to design and develop such a software environment. This is a huge undertaking and a great amount of effort is needed in order to achieve this task. Considering the time and resource limitation of this PhD project, the focus will be put on research, proposal, design and development of a new framework that integrates the identified and optimised functional and metadata structures. In this framework, various methods will be studied and developed to identify and optimise the required functional and metadata structures to support and manage both multi-disciplinary and inter-disciplinary simulation data and processes, embedded in a PLM environment.

## 1.2 AIM AND OBJECTIVE

The overall aim of the research project is formulated as follows:

**Researching, proposing, designing and developing a new framework that supports and manages multi- and inter-disciplinary system-simulations embedded in a PLM environment.**

The objectives are:

1. **Analysing and evaluating multi-disciplinary simulation data and their collection processes, the existing PLM software and their applications to identify and optimise the required functional and metadata structures to support and manage multi-disciplinary simulation data and processes embedded in a PLM environment.**
2. **Analysing and evaluating inter-disciplinary collaboration of various simulation software to identify and optimise the required functional and metadata structures to support and manage inter-disciplinary simulation data and processes embedded in a PLM environment.**
3. **Researching, proposing, designing and developing a new framework that integrates the identified and optimised functional and metadata structures to support and manage multi- and inter- disciplinary simulation in PLM system environment.**

The research questions are:

**What kinds of functional and metadata structures are required to support and manage multi-disciplinary simulation data and processes embedded in a PLM environment?**

**What kinds of functional and metadata structures are required to support inter-disciplinary collaboration of various simulation software embedded in a PLM environment?**

**Can these functional and metadata structures be integrated into a framework to support and manage multi-disciplinary and inter-disciplinary simulation in PLM system environment?**

## 1.3 DISCUSSION OF THE PROBLEMS

If a company is interested in the virtual test and validation of their products, simulation tools have to be employed. Most companies are interested in this because the time for a new



product to go to market and the development costs are important factors and risks can be reduced with the use of virtual development tools. As AberdeenGroup (2009) discussed:

“Iterative build and test cycles add additional cost and consume valuable time in the schedule. More importantly, designs are often too constrained late in the process to properly address problems identified during testing, forcing many to adopt meagre design compromises to get the product out the door.”

In order to reduce the amount of time to go to market yet improve the quality of products, companies are willing to optimise their development process. Companies could use simulation tools to test and validate their products and improve their development process. However, simulation tools are based on the underlying mathematics and numerical approaches (Dehning & Wolf, 2006, p. 8) and limited to their mathematical base. This means that simulation tools can solve only problems that can be described by their mathematical base. However, the validation of modern products often requires more than one mathematical base. Most products and processes include multi-physical systems where different physical disciplines work together.

The products and processes are often built as an amalgamation of different natural science disciplines (mechanics, electronics, controls etc.). For product testing and validation, the use of different natural and engineering science disciplines also requires the cooperation of different natural science and engineering disciplines described through varying mathematical forms. This could require multiple and different simulation tools to describe systems. Monolithic and even integrated simulation tools will often not be useable because these simulation tools are limited to their mathematical base. Vadim Engelson (2000) wrote in his dissertation:

“Simulation is typically used to optimise product properties and to reduce product development cost and time to market. Whereas in the past it was considered sufficient to simulate subsystems separately, the current trend is to simulate increasingly complex physical systems composed of subsystems from multiple domains such as mechanical electric, hydraulic, thermodynamic, and control system components.”

So, whilst companies are interested in simulating their products and processes, it also means they have to simulate these products and processes as a combination of the natural science disciplines that are involved in the system (Arnold & Schierz, 2009). In such a case, a multi-disciplinary view on the system is required. The virtual testing and validation of such a multi-disciplinary system could require multi-disciplinary simulation bases. Different simulation tools are required to generate simulation models based on multi-disciplinary simulation bases. A simulation tool can simulate a specific discipline or maybe only a part of this discipline. Therefore, the different disciplines and their simulations have to be linked to each other (Arnold & Schierz, 2009). This gap between simulation tools has to be closed if the complete functionality of the product is to be simulated. In the book “Why do Multi-Physics Analysis?” (Dehning & Wolf, 2006), this is summarised in one sentence:



“[... ] multi-physics applications are indeed clearly more complex than mono-physics simulations, but they are essential in many cases in order to obtain adequate engineering understanding.”

The difference between mono- and multi-physics simulations is not based on the combination of the numbers of used natural and engineering science disciplines. According to Dehning & Wolf, a simulation generation based on one simulation application is a mono-physics simulation and a simulation generation based on more than one simulation tool in couple mode is a multi-physics simulation.

In order to meet the simulation requirements of companies, simulation application vendors changed their technologies. First the simulation software was oriented as mono disciplinary simulation application (Arnold & Schierz, 2009). However, with the ongoing development of the simulation products, some of them were improved based on more than one mathematical approach. With one simulation product, the user can address different disciplines of simulations such as combined thermal and flow simulation applications. This does not mean that the solving technology will be combined into monolithic application approaches. The most commercial simulation products are a combination of multiple simulation applications. Different simulations but combined simulation applications could be coupled internally in the simulation product. However, there will be borderlines. Nevertheless, the internal coupling would be the most effective one. It would also be possible to couple externally. This could achieve the coupling of simulation tools from different vendors.

Different commercial products will be available on the market to link and couple simulation tools (Otter & Elmqvist, 1995, S. 2). Most of them will fix special problems whilst some try to be more universal. All of them use interface technologies for communication between different applications. These interfaces will transfer information bidirectionally or unidirectionally from and to a simulation tool which requires an input port for the simulation tools to receive data. However, the interfaces will mostly be individual and won't support a high number of simulation tools. In recent years, several standard interfaces have been created such as the MODELISAR – Interface (Consortium, October 12, 2010). Unfortunately, they are not generalised and are available only in a small, though growing, number of simulation tools (Consortium, 2011).

An open and general support of interacting simulation tools have to be independent from the art of interface (Kossel, Claudene, & Loeffler, 2009). This kind of application should be more than an interface. For example, multiple simulation tools, linked together for co-simulation, require that the simulation system runs in convergence manner (Dehning & Wolf, 2006, pp. 10-11). The convergence could fail because the convergence criteria will not be achieved. A convergence criterion can be 'energy balance' or 'force balance'. In the case of unachieved convergence, the simulation time step size has to be repeated or reduced. Thus, the convergence control requires sampling and time management and possibly, error correction. This could be achieved by using simulation middleware such as the software application called ICOS. This software application will be provided by the international research centre Virtual Vehicle in Graz (Zehetner, Wenpu Lu, Watzenig, & Bernasch, 2012)

A mono-physic simulation does not require this complexity of data and tools, or tool architecture. In contrast to mono-physic simulation, the data management of multi-physics simulation will be much more complex. Multi-physical simulation requires multiple simulation tools as well as multi-physic simulation architecture designs. The simulation models, their sources and the simulation results have to be managed. Thereby, the simulation can only be as good as the input of the simulation. So, the simulation model sources will play an important role in securing the simulation result quality and requires to be linked to the simulation. In the case of multi-disciplinary-simulation with multiple input sources, each simulation will require specific sources (Kübler, 2000, S. 21). In order to keep the sources and the simulation data of multi-disciplinary simulation transparent, this data should be managed.

PDM-systems (product data management) are normally used as the technology for the management of engineering product information and data. Engineering product information could become the most important source for the simulation models because the descriptions of the product would usually be the most important input (such as CAD models) for simulation. Commercial technologies and solutions to managing a single simulation model will need to relate simulation data to the simulation sources. TEAMCENTER UNIVIED from SIEMENS INDUSTRY SOFTWARE GmbH & Co KG will provide one of the leading solutions. Nevertheless, these technologies will not provide a solution for supporting and managing multi-disciplinary simulation data in an effective way. In Section 1.4, the improvement of the multi-disciplinary simulation data and process-management will be discussed.

#### 1.4 THE DISCUSSION OF THE IDEA OF A NEW FRAMEWORK FOR SUPPORTING AND MANAGING MULTI-DISCIPLINARY SIMULATIONS

The idea is to conceptualise a new framework that supports and manages multi-disciplinary simulation. This new framework should support the simulation from the task of the generation of a single simulation model to the task of obtaining analysis results from the multi-disciplinary simulation. The multi simulation models could be created by different departments and possibly by different companies. The individual simulation models will be connected and linked into a multi-disciplinary simulation model. Therefore, the individual simulation models have to be reusable and traceable in the environment of multi-disciplinary simulations. Additionally, the individual simulation models should be traceable to their sources. The traceability to their sources should also be achieved for multi-disciplinary simulations. The solve run (a solve run is the mathematical calculation of a simulation model) of multi-disciplinary simulations will generate multiple simulation results data that should be re-traceable as well.

Furthermore, the new framework should improve the support and management of multi-disciplinary simulation data. Additionally, the new framework should include the functions

for interacting directly with simulation source data storing and managing systems. In this thesis, PDM and PLM systems are used as the simulation source storing systems. So, the focus will be on conceptualizing a new framework that can be used to improve the support and management of multi-disciplinary simulation data embedded in a PLM environment.

Within a PLM environment, the generation of simulation models could be supported at an early stage of a product lifecycle. This data is recorded together with different metadata (Brendel & Kühner, 2004; Siemens Industry Software GmbH & Co.KG. , 2012). The data could be re-used in a later stage of the development process. At later stages, it is frequently required for the product design and development team to build a virtual prototype of product. This will reduce the product lead time (Mahler, 2012). In order to build the virtual prototype, the team has to identify and optimise the required simulation models. Thereby, they could use the simulation model metadata defined at the early stage. Then, they have to determine the processes for the simulation model data collection in both multi-disciplinary and inter-disciplinary environments. However, there is no existing software available to help a team in a PLM environment achieve this goal (Ai, Chen, Wan, & Xiong, 2011; Vyatkin, Hanisch, Cheng, & Chia-Han, 2009).

Therefore, it is not only theoretically meaningful, but also practically important to design and develop such a software environment. This is a huge task that requires a great amount of effort in order to fulfil this task. Considering the time and resource limitation of this PhD project, the focus will be on the research, proposal, design and development of a new framework that integrates the identified and optimised functional and metadata structures. In this framework, various methods will be studied and developed to identify and optimise the required functional and metadata structures to support and manage both multi-disciplinary and inter-disciplinary simulation data and processes embedded in a PLM environment.

Accordingly, the research project will evidently be influenced by the dependencies of the single and multi-disciplinary simulation models and the multiple simulation results. The focus will be on the data management.

## 1.5 ORGANISATION OF THE THESIS

Chapter 1 introduces the research and PhD thesis. The research aims and objectives, the requirement discussion plus the idea of a new framework to support and manage multi-disciplinary simulation in a PLM environment will be presented.

Chapter 2 presents the literature review of previous work in the areas that are related to this project topic. An insightful discussion into the areas of 'multi-disciplinary-simulation',

'simulation data management' and 'data management of multi-disciplinary simulation' will be given. Based on that, the gap between supporting and managing multi-disciplinary simulation, and the proposition of the new framework, will be discussed.

Chapter 3 presents the methodology of the research approach including discussions relating to the ontological approach, the methodology and methods used for this PhD project.

Chapter 4 presents the data collection, presentation and analysis of simulation supporting systems and technologies in a PLM system. The selection of the PLM system is also discussed by relating to the Chapter 2.

Chapter 5 presents four multi-disciplinary simulation case studies. These case studies discuss the support of data management and its processes. The data will be handled using a PLM system presented in Chapter 4.

Chapter 6 presents the new framework for improving the support and management of multi-disciplinary simulation in a PLM environment. The organisation of a system-simulation-structure, a system-simulation-result-structure and the collaboration of systems describing structures are discussed. Furthermore, an approach of combining system-simulation-structure and process-oriented simulation processes will be presented.

Chapter 7 presents a discussion of the new framework approach. A brief summary of the improvements, benefits and the verification of the new framework approach will be presented in the first subsection 7.1. Critiques and further work on the new approach will be presented in section 7.2. The chapter will end with the conclusion presented in the subsection 7.3.

## 2 LITERATURE REVIEW

### 2.1 INTRODUCTION

In Chapter 1 the project's scope and objectives have been presented. This chapter will presents the critical review of literature in supporting and managing multi-disciplinary system-simulations. The subtopics are shown in Figure 2-1.

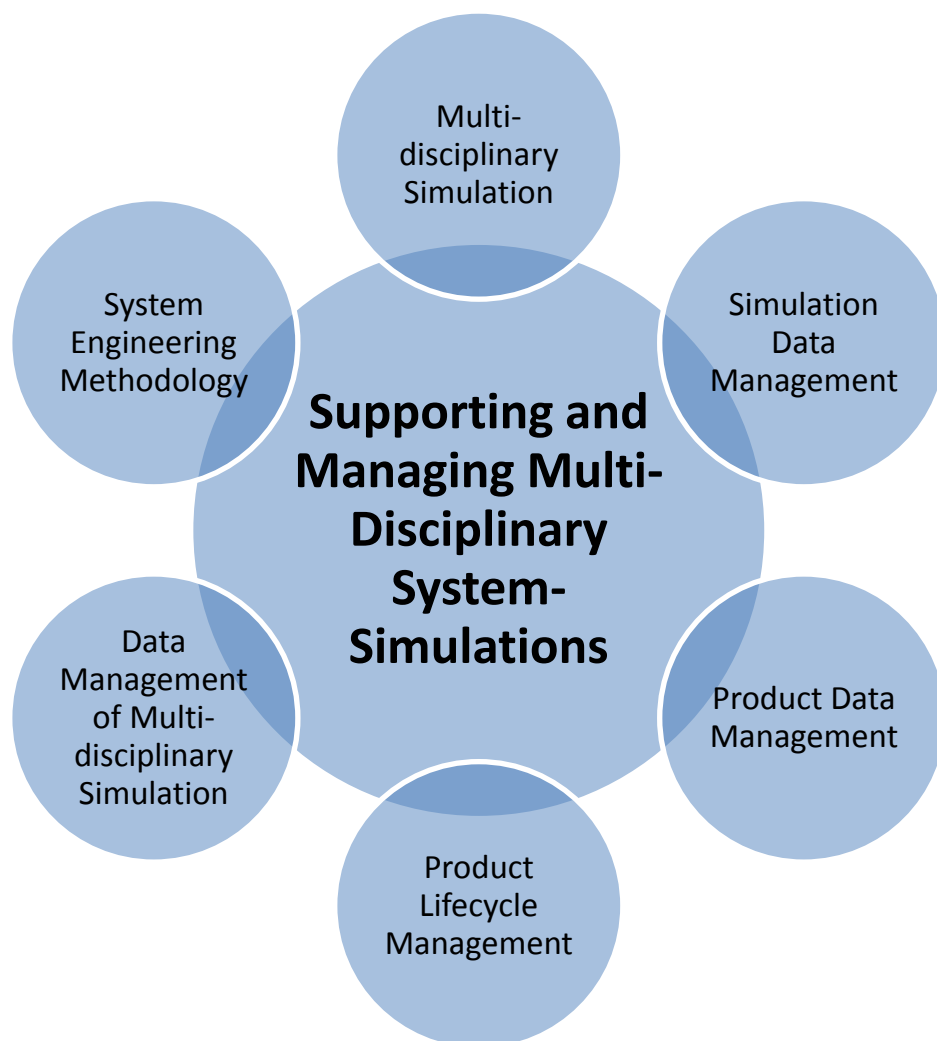


Figure 2-1 Research topic and subtopics

These sub-topics are:

- Multi-disciplinary simulation

- Simulation Data Management
- Product Data Management
- Product Lifecycle Management
- Data Management of Multi-disciplinary simulation
- System Engineering Methodology

## 2.2 MULTI-DISCIPLINARY SIMULATION

If a company is interested in using virtual prototypes to develop and validate their products, they usually use CAD models for design, and CAE simulation for validation and verification. Companies are interested in this because the time to go to market and the development costs are important factors and risks can be reduced with the use of virtual development tools such as simulation tools. Vadim Engelson (2000) thought that:

“Simulation is typically used to optimize product properties and to reduce product development cost and time to market. Whereas in the past it was considered sufficient to simulate subsystems separately, the current trend is to simulate increasingly complex physical systems composed of subsystems from multiple domains such as mechanical electric, hydraulic, thermodynamic, and control system components (p.125).”

Today, companies use simulation to optimise, validate and verify their products and processes. However, the simulation tools are based on the underlying mathematical and numerical approaches (Dehning & Wolf, 2006, p. 8) and limited to their bases, i.e. the simulation tools can solve only problems that can be described by their mathematical base. Modern product- or process- development often needs more than one mathematical base for its simulation. Most products and processes are multi-physical solutions where different physical disciplines work together. The products and processes are largely built as a mechatronic community including different disciplines (mechanic, electronic, control, etc.). Hui, Liping, Li & Tifan (2011) stated:

“Products are the complex systems with multiple disciplines, such as mechanical, electronic, hydraulic and control, whose creative development stems from single domain to multi-domain, from single application software to comprehensive application of several types of software.”

Single-disciplinary simulation is not able to maturely represent the product or process. Multiple disciplines of simulation have to be applied. The most industrial used simulation tools are specialised mono-disciplinary simulation tools (Arnold & Schierz, 2009). So, it is necessary to use multiple simulation tools. Unfortunately, the multiple simulation models of the same product or process, addressing different challenges (such as different kinds of abstract simulation based representations of products or processes), have to interact with

each other. A solution is to couple the independent mono-disciplinary simulations in parallel or serial. The coupling of different physical simulation components is called multi-physics analysis, for example thermodynamics is coupling thermal with fluid dynamics (Dehning & Wolf, 2006). The parallel coupling of two or more simulation tools is called co-simulation (Arnold & Schierz, 2009). Multi-physics analysis can be done with or without co-simulation. Dehning & Wolf (2006) stated:

“[...] multi-physics applications are indeed clearly more complex than mono-physics simulations, but they are essential in many cases in order to obtain adequate engineering understanding.”

Accordingly, multi-disciplinary simulation means multiple disciplines should be considered in integrated way. This kind of discipline is not defined (such as mechatronic, physical or simulation disciplines). All kinds of multi-physics simulation are summarised under the multi-disciplinary simulation. Multi-physics means the integration of coupling different physics simulations (such as thermal and electric).

There are different commercial solutions to link simulation models (Otter & Elmqvist, 1995, p. 2). Most of them fix a special problem and others try to be more universal. All of them use interfaces to transport information unidirectionally or bidirectionally between the simulation models. However, there is no interface that is generalised and supported by a high number of simulation tools. A standard interface could be a solution. However, such a standard interface for a more universal use to link different simulation models had to be independent (Kossel, Claudene, & Loeffler, 2009). In recent years, some companies pushed and tried to achieve a standard interface with the development of MODELISAR – Interface (Consortium, 2010).

The correspondence through these kinds of interfaces is usually dependent on the simulation time. Mostly, time-dependent multi-disciplinary simulations are needed to represent a time-dependent process or task. Only in some special cases is it possible to reduce the time-dependence of a simulation to a snapshot view such as an impact analysis of a maximum load case. In the case of a time-dependent simulation, also called transient simulation, the virtual time-line is subdivided into time samples. Time-sampling can be fixed or flexible, generated automatically or manually. Therefore, the virtual time-handling should be managed (Consortium, 2010, S. 5). Sometimes the convergence of the multi-disciplinary simulation is not given. This can be caused by the instability of the energy or force balance. An automatic correction of the virtual time-sampling step size could improve the convergence of the multi-disciplinary simulation (Dehning & Wolf, 2006, pp. 10-11).

The technological aspects of multi-disciplinary simulation, described above, are critical. A simulation middleware organises the interface-technology, the virtual simulation time-handling and the convergence control. Another simulation middleware organises the communication between the different simulation models by using interface technology. Thus, simulation middleware runs the simulation models, deciding the simulation time step size by controlling the convergence criteria. One simulation middleware example is an extracted software application such as ICOS from Virtual Vehicle in Graz, Austria. Alternatively a



simulation application can even own functionalities to run the task of a simulation discipline and the task of a simulation middleware. Such a simulation and simulation middleware application will be NX Motion.

In addition to the simulation architecture, there are base boundaries to achieve a realistic simulation: A simulation can only be as good as the input of the simulation. In the case of multi-disciplinary simulations, there are multiple simulation inputs (Kübler, 2000, S. 21). Two or more single simulation models are an input for the multi-disciplinary simulation. Additionally, a single simulation also requires input data. All this data has to be available; missing data reduces the simulation result quality.

All the previously discussed boundaries, influences and knowledge will be required for a multi-disciplinary simulation. Losing data or information will reduce the achievable quality of the multi-disciplinary simulation result. In order to secure and keep the data and information, the storage and management of this data and information is required. Simulation data management, discussed in Section 2.3, aims to meet such requirements.

## 2.3 SIMULATION DATA MANAGEMENT

This section reviews the management of simulation data and the support and management of simulation processes. The review will focus on the management of single simulation data and processes rather than the management of multi-disciplinary simulation data. As mentioned in Section 2.2 the data of single simulation models have to be re-used in the case of multi-disciplinary simulation. So the handling of single simulation data and information has to be respected in the multi-disciplinary context.

The management of knowledge and data, that is used or generated by simulation processes, can help improve the development process. The efficiency and economy of simulations, the design and resource investment into simulation and design are cumulated. Jenkins (2012) mentions:

“[...] Interviewees identified insufficient focus on knowledge capture, data sharing and re-use as major constraints on the value available from simulation and analysis. Data needed by collateral and downstream project functions is too often unavailable, outdated, or captive to error-prone manual methods of dissemination and re-entry. Also needed are ways to capture and share best practice work processes beyond the project where they originated. [...] Individual analysts and engineers reported pragmatic needs for solutions that let them work faster and with higher confidence. [...] Beyond this [...] most would find simulation data and process management of greatest value [...]”.

The study of Jenkins (2012) was based on “some two dozen program managers, discipline leads, analysts, engineers and other employed [...] in North America, Europe and Asia.” The interviews were focused on the “best practices [...] for implementing digital simulation and analysis and maximizing its business impact.” This research was carried out across a range of



manufacturing industries including aerospace and defence, aircraft engines, automotive power trains, consumer electronics, medical devices and off-highway equipment. Jenkins (2012) thought that it would be necessary for the SDM/SPM (Simulation Data Management/Simulation Process Management) to be “embedded in context to the product structure management.” Instead of the singular view on SDM and SPM, a merged and common view was given about SDPM (simulation data and process management).

Mostly SDPM will be used to manage single simulation models. So, simulation is bonded to a huge amount of data and information and a high number of files. This can be best demonstrated in a simplified example of standard FEA (Finite Element Analysis) simulation process Heber & Gray, (2005):

1. There should be a description of task and input-data such as CAD models, load cases and boundaries etc., for the simulation to provide the information an analyst needs about what is to be examined. This information has to be generated by the person in charge. → The description information is stored in documental formats such as MS Word and MS Outlook etc.
2. The data for describing models has to be generated. This kind of data is predefined from the product development department in a CAD format. In that case, the data will be mentioned in the description of the task. → In most cases, this will be stored in CAD formats such as NX and Solid Edge etc.
3. The data for representing models has to be abstracted in models that are ideal for meshing or simulating. For example, an engine bonnet is abstracted to the mid-surface and the thickness. This process is the first part of a so-called Pre-Process. → The abstracted data is usually stored in initiative CAD formats as well, but they could also be stored in CAE formats such as FEMAP and ANSYS WORKBENCH etc.
4. The simulation expert has to transform this abstracted model to a finite element (FEA) based format. This means that the abstracted CAD model has to be abstracted to a mesh with FEA-mesh-specific descriptions. “The meshing phase decomposes the model geometry into simple shapes or voxels like tetrahedral or bricks that fill the volume (Heber & Gray, 2005)”. This process is part of the Pre-Process and done in a FEA-Pre-Process-tool. → This mesh data will be mostly stored in initiative FEA-Post-Process-tool-specific files such as NX, FEMAP and ANSYS WORKBENCH etc.
5. Based on the generated mesh or meshes, load cases and boundaries have to be added. A load case describing force and moment influences on mesh nodes and boundaries is called a mesh-node-freedom-degree-reduction. The FEA-tool adds the loads and boundaries. Specification of solver parameters has to be filled out and defined. “The finite element analysis is an approach to modelling partial differential equations by

replacing the continuum problem with an approximate discrete problem suitable for numerical solution in a computer (Heber & Gray, 2005).” A solver takes care and automates the mathematical routines. → These load cases, boundaries and solver parameters will be mostly stored in initiative FEA-Post-Process-tool-specific files such as NX, FEMAP, ANSYS WORKBENCH etc.

6. This file is not readable by a FEA-solver-tool. A FEA-solver needs a special file format. The file format is dependent on the solver. A solver-specific file is generated by the FEA-Post-Process-tool (Inc., NX Nastran User's Guide, 2009). This file is called input-deck. → An FEA-solver-specific-input-deck file is generated by the Post-Process tool readable by solvers such as NASTRAN and ANSYS etc.
7. An FEA-solver, solving the mathematical routines, loads the input-deck file. Based on the data included in the input-deck and the automated mathematical routines, the solver will generate matrices. Most of these matrices are only saved temporarily, but in some cases, it will be necessary to keep this data. Afterwards, the matrices are solved with the automated mathematical routines of the solver. The results will be stored in result files. The number of result files and their formats will be dependent on the FEA-solver and the defined solver parameters. There are files including results and others describing the FEA-solver routines and work, for examples log-protocols of the routines, warnings and errors. The log-files should make the solve-process reviewable (Inc., NX Nastran User's Guide, 2009). → The solve run will generate user-predefined result files, protocol files and output files in the FEA-solver specific formats such as NASTRAN and ANSYS etc.
8. The result-files include result data like stress or deformation. Unfortunately, the result files are very big tables and are not suitable to post simulation analysis. With Post-Processor tools, the results can be visualised using a virtual environment instead of tables. Based on this visualisation, the results become interpretable by an analyst. The output files and protocol files could be opened with text-reading tools such as Notepad to get additional information. Besides, the use of Post-Processing-tools requires specific user know-how. Additionally, the analysis and the interpretation of the simulation are not documented by the Post-Process tools (Heber & Gray, 2005). To document the interpreted results, other tools such as Microsoft Office are preferred. In order to better understand the result interpretation, two or three dimensional images are integrated in the results documentation. Therefore, Post-Processing tools provide the possibility to derivate images such as two dimensional jpg-, gif-formats or three dimensional JT-formats or videos such as avi-format. These images should be viewable with popular tools such as Microsoft Office or PDF readers or JT-Viewer such as JT2GO. → The interpretation of the results should always be documented. The documentation is stored in documentation files such as MS Office documents and picture and movie files such as jpg-, gif-, avi- and JT-format.

The example in Section 2.3 is a standard simulation example and describes the required and produced data of a single simulation during the simulation process. In the case of multi-disciplinary simulations, the amount of data will be multiplied. Each of the simulation models will be generated during its own simulation process. Afterwards, the individual simulation models will be merged into the multi-disciplinary simulation. A large number of files and information are generated during multi-disciplinary simulations, much bigger than the number of files and information in a single/individual simulation case. With the growing number of required and generated data and files, the use of simulation-data-management will become more important. Special and commercial applications for managing multi-disciplinary system-simulation data will be available, but they will not address the support and management of the described single simulation process. Such special and commercial applications will be discussed in Section 2.6.

However, a single simulation requires traceability to its simulation sources. A traceable management of the data and information guarantees the quality of simulation (Ulrich, 2011, p. 56). Two main issues should be solved for data traceability and this can be achieved using a simulation data management tool:

- The design development continues and creates new versions of the product, mostly in the form of CAD models, whereas the simulation department works with an old version of the product-representing CAD models. Relationships between the data should provide a possibility to check the actuality of the simulation against the actuality of the designed product descriptions (VDA, 2008).
- In the case of freezing a simulation, i.e. the simulation and analysis ends with a positive result or a necessary design change, the bases of the simulation have to be traceable individually. Therefore, the simulation and the simulation source data, such as CAD models, have to be linked to each other. This means the simulation has to be linked to the source versions. The used base data of the simulation, such as CAD models, is called the base line of the simulation. This base line has to be traceable. The SDM tools have to take care of product data changes, such as CAD models or documentations, during an on-going development or change process after the generation and freezing of simulation data and information (VDA, 2008).

Simulation files should be managed by simulation data management tools (Sebastien & Ducellier, 2006). In addition, metadata such as the author of a simulation, the date of creation or data change, link-paths to external data and the state of the simulation, etc., should be managed by SDPM (VDA, 2008). The metadata information can be helpful, for example, in identifying the person responsible for the simulation model or the interpretation of a simulation result, and the date the data and information was generated. Metadata is necessary to give answers to questions like 'Who has made what, when and why?' or 'What dependencies are given for this information?' (Boy, Grau, & Trautmann, 2010). Knowing such information will help to make decisions like 'Is the simulation too old for re-use?' or 'Who could update the simulation?' Such decisions would be made during a product development or product change process.

Simulations could have a high impact on other areas in the product lifecycle (Sebastien & Ducellier, 2006). Concepts, designs, manufacturing processes, transportation and usage of the products could be optimised with simulation. This is because simulations can represent a challenge, product or scenario. For detailed and precise interpretation the simulation should be as realistic as possible. This requires a large amount of information and knowledge about the addressed challenge, product or scenario. Product data management (PDM) or product lifecycle management (PLM) could manage this required data and information. In order to merge such information with the data of the simulation, managed by SDPM, it will be useful to merge the data management approaches of SDPM with PDM and/or PLM (Boy, Grau, & Trautmann, 2010). The PDM approach will be discussed in Section 2.4 and the PLM approach in Section 2.5.

A simulation process can be complex and extensive. Moreover, if the simulation process has to be documented or reproducible, it also makes sense to manage the simulation process itself. If the simulation process is fully or partially automated, costings and timings will be reduced if the process has to be repeated multiple times. So, the management of simulation processes could provide the required data, information and metadata. Fachbach & Rosenberger (2010) consider that simulation process management, in addition to simulation data management, could:

- build a base for automation.

Simulation could be realised with a higher efficiency by automating the simulation process that generates the simulation. Such automation will help to achieve a constant quality in the generation of simulation models. Additionally, the management of the simulation process will also generate documentation for later traceability.

- build a base for collaboration.

Different tasks could be necessary in a simulation process. The tasks could be undertaken by different persons and departments. Managed simulation processes could support the individuals taking part in the process. Therefore, the management of the simulation process could assist with 'push and pull' of data and information. This could improve the efficiency of simulation tasks. For the 'push and pull' or checking of data and information managed in a SDPM or PLM system, a collaborating simulation process management has to be integrated into these systems.

The opinions of Fachbach & Rosenberger (2010) will be valid for individual/single and multi-disciplinary simulation models and processes. Fachbach & Rosenberger (2010) state:

"SDPM concerns CAE data, processes, methods and maybe resources. These elements are components of more complex data flows and overall processes. SDPM can therefore be considered not as an isolated system – rather, it plays the role of a central middleware for a variety of processes and systems. [...] Tight integration with other data management

systems and all decision-making processes, as well as the wide range of different tools in the field of CAE is the elementary challenge [...].”

The ultimate goal of SDPM is to provide an efficient decision-making data base including simulation data and processes. A system, product or part of a product or system has to fulfil functionalities. Each function has to fulfil technical requirements. Fulfilment of the requirements could be used for validation and verification of the system, product or subsystems and subproducts. The achievement of requirements could be checked by simulations (see also Section 2.7), which will be useful for the decision-making process. SDPM will help to manage data, information and processes to make them useable in context with other data and processes. Fachbach & Rosenberger (2010) report:

“Data supply for the virtual development process has to be guaranteed and the gap between the different systems has to be closed in a traceable way. The focus will be on the support of the decision and release process.”

The viewpoint of SDPM will also be included in the viewpoint of simulation lifecycle management (SLM). CIMdata Inc. (2011) elucidated that SLM (simulation lifecycle management):

“is to transform simulation from a specialty operation to an enterprise product development enabler that spans many segments of the product lifecycle. To do this, SLM should provide technology in four foundational areas: simulation and test management, simulation and test process management, decision support, enterprise collaboration”

In this thesis, SLM has additional functionalities to SDPM such as decision support and enterprise collaboration.

CIMdata Inc. (2011) mentioned special functionalities of a SLM system:

- SLM should support decision processes:  
“Fundamentally, organizations perform simulations to validate decision making based on functional, logical, and physical requirements. SLM provides capabilities to capture and present simulation information and results to enable these design decisions.”
- SLM should support enterprise collaborations:  
“With SLM, simulation is no longer decoupled from the product lifecycle. Simulation data and processes can be linked with requirements, parts, the BOM (bill of material) and other elements in the PLM process. Verification and validation of the design becomes more than a check in a box. Approvals have more substance than just a signature. Users can navigate to the exact simulation results that drove the design decision. SLM helps make it straightforward to see the genesis of the design – why certain designs were selected in favour of others. This exposure of simulation to the enterprise PLM provides a critical bridge between design and engineering.”

However, in the commercial market, the derivations based on such functionalities will not be made. Though the commercial software approaches are called SLM, SDPM and SDM and are

independent of the functionalities they have, the functionalities of commercial applications and academic oriented approach descriptions (SDPM, SDM, SLM) are incompatible. An SDPM tool could also be an SLM tool, such as TEAMCENTER.

Nevertheless, SLM seems to be a combination of SDPM, PDM and PLM. So the combination of SDPM and PDM or PLM could be a way of achieving improvements in multiple cases (Sebastien & Ducellier, 2006). Therefore, Section 2.4 will discuss the combination of PDM and SDPM, and Section 2.5 will discuss the combination of PLM and SDPM.

## 2.4 PRODUCT DATA MANAGEMENT

Because PDM could be a success factor, companies are interested in the organization and collection of product development information in one database. VDA (2008) stated:

“An increasing number of companies have introduced and are using product data management systems [...] to control engineering data storage and to manage engineering workflows. [...]”

This will help to organise multiple development departments and multiple experts (Brendel & Kühner, 2004, p. 62). All those departments and experts are dependent upon each other because their work is dependent upon each other's information and data. A single database holding this information will facilitate the search for relevant data and information related to the individual work (Sebastien & Ducellier, 2006). So, PDM systems, also called EDM systems (Engineering-Data-Management) (Cummings, 2006), are applied. Besides, regional gaps between the different individuals working on one project or product will be bridgeable through PDM systems.

Cummings (2006) mentioned that PDM still remains relevant today, for three key reasons:

- its ability to search for information using metadata,
- to manage interrelationships or virtual information,
- and to foster collaboration across geographical areas

### I. Search for Information

All data and information of a product and engineering project should be stored in a PDM system. Thus the PDM system should include all information and data that will present or represent the product and, as such, will summarise the product definition (Wikipedia Produktdatenmanagement, 2011) (Albers, 2011). Additionally, the metadata and information should be organised in the PDM system. Cummings (2006) described:

“Any valuable information that describes an attribute of the product can be classified as metadata. For example, search criterion could encompass product name, part or client name, or revision date. This allows searching and retrieval to be extremely focused [...]”

Such data is managed and provided by a PDM system. The PDM system also provides search engines to filter this data. Cummings (2006) mentioned that search functionality is important in engineering. Search functionalities, such as full text search for an overall search and metadata search for a specific and focused search will be required.

This data and information could include interrelationships and virtual information about the product. Data, metadata and information should interact with each other. Therefore PDM has to support both the management of interrelationships and virtual information.

## II. The Management Interrelationships between Information and Data

Interrelationships could occur between two- or three-dimensional CAD models. The CAD models represent the product design in a virtual environment. Interrelationships of the CAD models relate to objects such as drawings or even other CAD models. The interrelationships can combine or/and form different data representing parts or assemblies into a community such as system or product. Cummings (2006) mentioned:

“This interrelationship [...] is a ‘virtual construct’. This means that the only way that you know a particular screw or a bolt is related to a nut, is because there is this virtual construct called an ‘interrelationship’ between the two.”

Interrelationships can also be used to link different kinds of information and data. Dependencies between different information and data, such as engineering and design information, will become manageable. For example, engineering information could be from a Microsoft Word document describing requirements for a system or product (mostly generated before the design process begins) and design information such as a CAD model describing the developed virtual product (generated during the design process).

In addition to the improved traceability of the data and information, interrelationships will provide possibilities for a more complex search. This will foster the collaboration between different departments and persons taking part in a product development or change process.

## III. Foster Collaboration

In engineering departments, experts focus on different and specific software applications. It does not make sense to train experts in each engineering software application in a company yet not have access to all engineering software applications. Besides, it is necessary for experts to be able to access information that will normally be stored in files with restricted



access to the authoring software application. Such information could be synchronised with publicised metadata. Cummings (2006) noted that the PLM system should “[...] provide the ability for engineers to access that data without needing the [...] application installed. [...] They will be able to access, view and manipulate [...]” the data.

Access to data has to be restricted by managing access rights of users and the access to the data should be opened up for different localities. Not every user has the right to change design models or to view them. For example, engineers who are planning a production line should not be allowed to change the design, but they need access to view the CAD models. Cummings (2006) mentioned:

“EDM/PDM systems also provide a platform through which geographically remote teams can work on product design and engineering together. It provides the backbone and cohesion for teams to get accurate data to make informed business decisions about the products they design. [...] The cross-geographical capabilities of EDM/PDM systems make these solutions more relevant because it’s an absolute business requirement [...]”

The collaboration between departments cannot end at the boundaries of a PDM system. Simulation experts also require data and information from design and planning departments. The combination of SDPM systems used by simulation departments and the PDM systems used by designing and planning departments could foster this collaboration thereby reducing gaps in the simulation process.

#### IV. Combination of SDPM with PDM

SDPM and PDM are often separate systems. This is in contrast to the interacting and collaborating requirements between departments. SDPM, as a separate system, cannot achieve full interaction with other departments, tasks and projects in a company (see Figure 2-2). However, the interaction and collaboration between design and simulation engineering departments is important. So those companies investing in SDPM would also invest in collaborations between the SDPM system and other data management systems. Fachbach & Rosenberger(2010) mentioned that:

“[...] SDMPM couldn’t be seen as isolated system - it has to be more a central linking-system of a high number of processes and systems. [...] The tight integration with other data management systems and all decision-making processes and the wide range of different tools in the field of CAE is the elementary challenge of the implementation.”



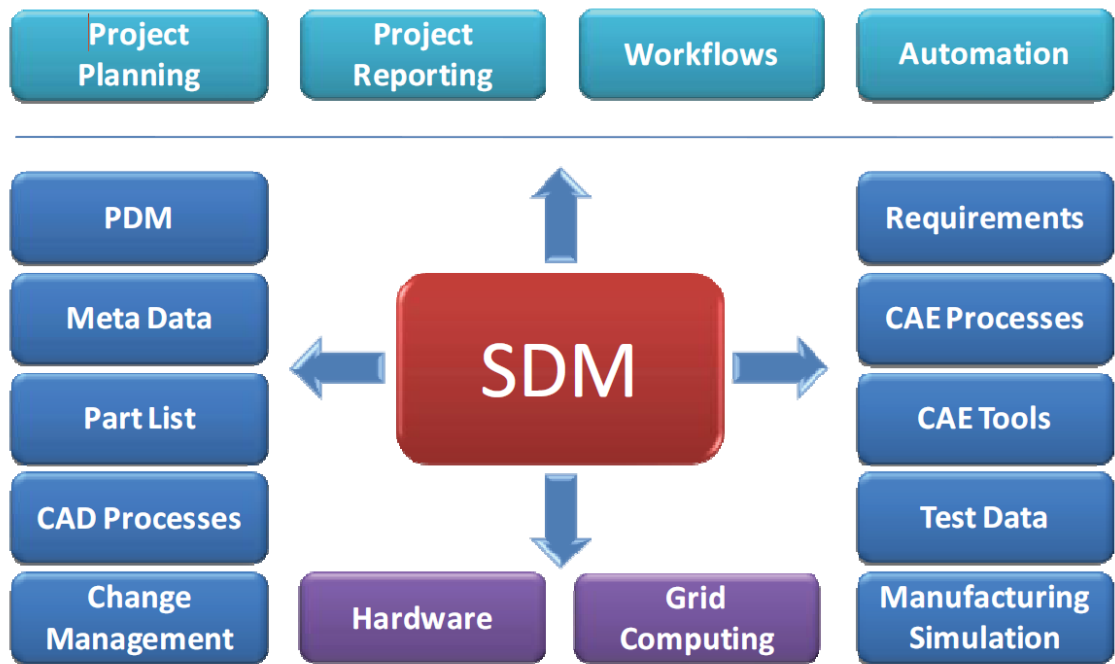


Figure 2-2 SDPM Integration in a Complex System- and Process-Landscape (Source: Virtual Vehicle)

Consequently, the combination of SDPM and PDM could foster collaboration and improve the efficiency and quality of engineering. VDA (2008) noticed:

“To increase the efficiency of virtual product validation and decrease the time to market, companies are striving to improve the synchronization and integration of the technical (CAD) engineering process and the virtual product validation process (simulation and computation). This implies an integration of simulation and computation data in the PDM environment.”

Fachbach & Rosenberger (2010) mentioned that:

“A major task of SDM is the link between CAD and CAE. This gap must be closed in such a way that simulation results and decisions based on the functional properties in a project can be traced at any time and from any perspective.”

A common collaboration framework between different data and information management tools used by the different departments, tasks and projects in a company will foster the collaboration and improve product quality. PLM tries to provide such a common collaboration framework as discussed in Section 2.5.

## 2.5 PRODUCT LIFECYCLE MANAGEMENT

This section introduces product lifecycle management. The review will discuss the stage of technology and different application approaches. This knowledge will be used for subsequent identification of the most important SDPM and PLM application in Section 4.

A PDM database will be only one component used to manage engineering knowledge. The lifecycle of a product has to go through different processes such as:

- Idea management
- Project planning
- Product development
- Product simulation
- Product testing
- Factory planning
- Production planning
- Controlling
- Sales
- Product Support
- Product Recycling

Since the number of departments, tasks, processes and time periods is great, it follows that the lifecycle of a product should be assisted by product lifecycle management systems. Robust and complex PLM systems can be used to provide information to management before, during and after development processes (Schuh, 2011). With large amounts of data, it makes sense that processes are managed and trigger-automated by the PLM system; they could also include automated checks such as availability of data or quality of CAD models. The predefining of processes and automation could reduce administrative work, increasing both quality (of the product and its processes) and efficiency. Therefore, the core technology of PLM systems will also encompass engineering data and management information through an interacting or mostly integrated PDM system.

“The key to any successful PLM initiative is at its core the EDM/PDM system. In fact, PLM does not exist in any shape or form without the EDM/PDM component to serve up accurate design/engineering data” (Cummings, 2006).

The focal point of PLM is the integration of distributed company locations and departments. Processes, data and IT systems should be integrated via a PLM infrastructure. Abramovici & Schulte (2005) discussed the vision of a continuous and company-location-overlapping process chain. One of the points that should be optimised in the future is the reduction of the number of data management systems. Multiple data management systems are a result of using multiple engineering software tools like CAE, CAM (Computer aided manufacturing) and DMU (Digital Mock Up) tools. Each of these systems or tools has its own requirements in combination with data management systems. There are also increasing application systems

such as CAx (computational added and x stands for the different options) in those areas where competitive advantage could be gained. These applications will generate special, additional data and information, which makes special and additional data management systems necessary. The individually specialised data management systems usually do not focus on general data management. So PLM tries to provide a common framework for all this data holding or data authoring applications.

The backbone of PLM is to integrate various concepts, methods and IT-tools into one data and information management system. This system should provide engineering information, processes and challenges in a co-operational, global and distributed product lifetime cycle to optimise the product from the viewpoint of a producer, customer and deliverer. All released data in the product lifecycle should be integrated into the PLM-tool and should be stored and managed in a central or multi-central archive (Abramovici & Schulte, 2006).

PLM requires deep integration of PDM (Abramovici & Schulte, 2006) and should build a common framework to integrate multiple data and information holding or producing applications. If SDPM is also integrated into PLM, the common framework of a PLM system will be further improved and the representation of a product lifecycle more realistic.

#### I. Combination of SDM with PLM

A greater amount of data and information of the product lifecycle will be available in a PLM system. Simulation requires detailed information and data about the product or system. More information and data could be used to carry out more realistic simulations. A more realistic simulation approach will improve the simulation quality because simulations should represent a challenge, product or scenario as realistically as possible. Therefore, it is necessary to integrate all influences appearing throughout the product lifecycle into the simulation (Boy, Grau, & Trautmann, 2010). A typical example is that loads or boundaries (like material properties etc.) should be considered based on resultant data of production processes. This data will also be stored and managed in a product lifecycle system (Ulrich, 2011, p. 56). An example of this is cupping of sheets which influences the material behaviour of the sheet. If this sheet is embedded in a car crash simulation, the material behaviour will not be congruent to the general material. The simulation expert should utilise the changed material behaviour after the cupping (Fachbach & Rosenberger, 2010).

In summary, PDM and PLM could include data and information that is useful for simulations. PLM includes a greater amount of data and information than PDM because more data and information in the product lifecycle is managed through PLM. In order to access more data and information about the product lifecycle and to improve simulation based on this data and information, an integration of SDPM into a PDM or PLM system will be useful (Hui, Liping, Li & Tifan, 2011). This integration could provide access to more important simulation source data and information of other departments, processes etc. (Giptner, Moshhammer, & Panzer,

2010, p. 222). So, the simulation is both high quality and bears a much closer resemblance to physical reality.

Most commercial SDPM tools are stand alone applications and only provide interfaces to PDM or PLM systems. Only a few of them are integrated into PLM systems. In the following sections, commercial SDPM tools from five leading vendors will be evaluated from the viewpoint of their interactions with PLM approaches.

## II. SDPM Application Interaction with PLM Approaches of the Five Leading Vendors

Vrinat (2009) analysed the five leading vendors of SDPM in his research report. He identified them as:

- 1) Company: ANSYS  
Product: EKM (Engineering Knowledge Manager) and Workbench V10 (Released Q2 '09)
- 2) Company: MSC Software (MSC)  
Product: SimManager and SimXpert R14 (Released Q2 '09)
- 3) Company: Siemens PLM Software (Siemens)  
Product: TEAMCENTER for Simulation V8 (Released Q3 '09)
- 4) Company: Dassault Systèmes / SIMULIA (DS/SIMULIA)  
Product: Enovia SLM V6 R2010 (Released Q2 '09)
- 5) Company: Altair  
Product: Hyper Works Enterprise (Released Q4 2009/Q1 2010)

The same leading vendors are also listed in Ulrich (2011). These five vendors actually provide the most effective simulation data management tools.

In relation to the support and management of multi-disciplinary system-simulation in a PLM environment, the expectation and cooperation on a SDPM and PLM interaction is high. So, it is not only single simulation processes and data that have to be managed and supported, but also merging multi-disciplinary system-simulations. The SDPM tools that are the best for cooperating and interacting with PDM and PLM applications could provide both important and valuable information for the research project.

All major vendors have achieved significant progress through data integration between different simulation authoring tools. In PLM integration, ANSYS and Siemens have achieved good ratings whereas all other vendor's ratings were relatively low (Vrinat, 2009). According to ANSYS (2011), ANSYS EKM has an interface to PLM/PDM applications such as Windchill from PTC and TEAMCENTER from Siemens. However, there is no document declaring that ANSYS EKM is integrated into a PDM or PLM system. This means that all vendors except for Siemens have to use interfaces to PDM or PLM systems for data integration. Data and

information that is not supported by an interface have to be handled using a PDM or PLM system separately from the SDPM system.

DS/Simulia or SIMULIA SLM from Dassault Systèmes is based on Dassault Systèmes's V6 platform technology. This solution includes scenario definitions, execution engines, live simulation reviews, Isight and other add-on components. The scenario definition is based on Dassault Systèmes ENOVIA V6 data management technology (CIMdata Inc., 2011). This combination represents a good but not full integration into a PLM system.

TEAMCENTER Unified ® from Siemens is a high level PLM and PDM system which includes a fully embedded SDPM tool. TEAMCENTER for Simulation, a specialised application on top of TEAMCENTER Unified Technology, has the highest ranking and coverage in the scorecard of Vrinat (2009). The product TEAMCENTER for Simulation is more a branding; the official product name is TEAMCENTER Simulation Process Management ® (PLM, 2012). In the case of TEAMCENTER Simulation Process Management ®, the SDPM system is directly integrated into the PLM system TEAMCENTER. This integration also supports the direct use and cooperation with TEAMCENTER System Engineering & Requirements Management ® (Mahler, 2012). Interfaces from TEAMCENTER to 3<sup>rd</sup> party software tools, like DOORS, are also available (Inc., Siemens Product Lifecycle Management Software, 2001).

In summary, TEAMCENTER can be seen as the best SDPM tool in achieving a high level of interaction with PDM and PLM applications. However, it still does not provide the full integration/management of the data and information generated in multi-disciplinary simulations. This is the focus of the proposed new framework. Section 2.6 will discuss this in more detail.

## 2.6 DATA MANAGEMENT OF MULTI-DISCIPLINARY SIMULATIONS

This section will review the data management of multi-disciplinary simulations. Current SDPM systems will be evaluated against their performances of supporting and managing multi-disciplinary system-simulations. Additionally, existing special applications for system-simulation data management will be discussed. The knowledge gained from this evaluation will be used for the purpose of identifying the most important SDPM and PLM applications in Section 4.

Following the development of multi-disciplinary simulations, there has been an up growth in demand for data management tools. Commercial SDPM tools do not meet all these demands. Some are met by special applications to manage system-simulation data such as LMS.SysDM of the LMS Company. Such applications manage behaviour models of selected simulation authoring tools (Matlab / Simulink and LMS.Imagin Lab) (LMS International, 2012). However functions of data management fundamentals have not been met. Furthermore, the connectivity to PDM or PLM are not available. In addition, the openness for cooperating with

multiple simulation authoring tools is ineffective. As far as the author is aware, there are no tools that offer universal support of multi-disciplinary simulation data management.

In the context of SDPM systems, the support of multi-disciplinary simulations could be misunderstood.

- 1.) One understanding could be that the SDPM application should support multiple simulation authoring tools. This means that SDPM should be independent and support multiple simulation tools. The disciplines of the simulation tools should not have any influence on the SDPM systems. This independence is not given in every case (Fachbach & Rosenberger, 2010). So, Vrinat (2009) and Fachbach & Rosenberger (2010) identified the need to support multiple and disciplinary-independent simulation tools with SDPM tools. Wrongly or misleadingly, these are often called multi-disciplinary (Vrinat, 2009).
- 2.) Another understanding could be that data and information of a simulation, that combine multiple disciplines, needs to be managed. In the case of supporting multi-disciplinary simulations, the different simulation tools or simulation disciplines have to be coupled. The data and information of the coupled simulation should be managed and represented by an SDPM system (Giptner, Moshhammer, & Panzer, 2010, p. 222-223). The thesis of Giptner, Moshhammer, & Panzer (2010, p. 222-223) elucidated for a special solution of SIEMENS AG Austria that the importance of coupled simulation will increase. However, this kind of simulation would generate heavy dataflow between different departments and companies in the development process. In order to provide this dataflow, an SDPM system embedded into PDM is required.

The understanding of multi-disciplinary in the research project will be congruent to the second point. However, the second point also includes the first point because the management of data and information generated by a multi-disciplinary simulation also requires the management of data and information generated by multiple simulation authoring tools.

## I. SDPM Systems for Supporting Multi-Disciplinary-Simulations

The SDPM systems presented in Section 2.5 II will be used as a base for the evaluation. The usability of the five SDPM systems above in the multi-disciplinary simulation data and process management will be examined.

Dassault Systems provides the products SIMULIA Multiphysics Digital Lab and SIMULIA Co-Simulation Engine to support co-simulations. These products can be coupled with their PLM product ENOVIA. SIMULIA Multiphysics Digital Lab supports the multi-physics simulation with simulation tools of Dassault Systems like the Abaqus simulation tools or products of

involved partners (Systems, 2012) such as CD-Adapco (Dassault Systems, 2010). Dassault Systems (2010) explain that Abaqus and CD-Adapco can be coupled to carry out fluid-structure-integrated simulations by using SIMULIA Co-Simulation Engine. CD-Adapco specialises in simulation tools for thermal and flow simulation. Abaqus, which is part of the Dassault Systems Company, concentrates on simulation tools for structure simulations. In order to link these simulation tools, interfaces such as the MpCCI interface from Fraunhofer SCAI Institute are used. However, no documentation (CIMdata Inc., 2011; Dassault Systems, 2011; Dassault Systems, 2012) includes information about the fact that Dassault Systems tools manage multi-disciplinary simulation data and process. It seems therefore, that effective functionality for supporting and managing multi-disciplinary simulation data and process has not been integrated.

Ansys EKM could cooperate with PDM enterprise databases, but it is an independent tool with file-based storage for the data. Only the metadata is stored in the SDM system database (ANSYS), hence it is not integrated in a PLM context. As far as the author is aware, there are no publications about functionalities with Ansys EKM to manage multi-disciplinary simulation data and process (Ansys, Inc., 2011). Similarly, it seems that effective functionalities for supporting and managing multi-disciplinary simulation data and process have not been integrated.

With the product SimXpert, MSC provides a multi-disciplinary simulation tool as authoring tools to generate multi-disciplinary simulation models (MSC.Software Corporation, 2011). However and so far as the author is aware, there are no publications including information about the functionalities of MSC SimManager to support and manage multi-disciplinary simulation data and process (MSC.Software GmbH, 2012). Again, as above, there is no evidence to show that effective functionalities to support and manage multi-disciplinary simulation data and processes have been integrated.

Similarly, and as far as the author is aware, there are also no publications incorporating the functionalities of Altair Hyperworks to support and manage multi-disciplinary simulation data and process (Altair Hyperworks, 2012). So the same conclusion can be drawn that the effective functionalities to support and manage multi-disciplinary simulation data and process have not been integrated.

As for Siemens TEAMCENTER, unfortunately, the author has not been able to find any publications about the functionalities that fully support and manage multi disciplinary simulation data and process (Siemens Product Lifecycle Management Software Inc, 2012). There is some low level integration however. For example, different individual simulation models can be related to each other. The interaction of individual simulation models during an inter- and multi-disciplinairy simulation could be represented by such relation. In addition, TEAMCENTER provides technologies for managing CAE data and EBS (equation based simulation) data and TEAMCENTER provides special technologies to configure implementation and cooperation with external authoring software tools. On this way



different simulation authoring tools and different simulation individuals could be supported.

The evaluation shows that none of the five leading SDPM tools provides the functions to fully support and manage multi-disciplinary simulation data and process. However, the evaluation identified that among these five tools, two SDPM tools provide some basic functions for the data integration management. One system is provided by Dassault Systems Company and the second system by SIEMENS Industry Software GmbH & Co.KG. The Dassault Systems Company focuses on specialised simulation authoring tools, whilst the SIEMENS Industry Software GmbH & Co.KG system is open to integration with other simulation authoring tools. This difference is important because the latter system offers more openness of a new framework for supporting and managing multi-disciplinary simulation data and process.

Notwithstanding, commercial but specialised applications for system-simulation data management are available and these are discussed in the next section.

## II. Commercial System-Simulation Data Management Applications

Few software tools address the data management of system-simulation data. Those companies that require system-simulation data management tend to create and program their own solutions, such as the work of Bindick, Lange & Lund (2012) and Bauer, Stüber, Meller & Gruber (2012). The leading vendor of system-simulation data management is LMS with its product Imagine.Lab SysDM.

“LMS Imagine.Lab SysDM is the solution to manage system data originating from LMS Imagine.Lab AMESim and other system-simulation tools, providing a collaborative environment for Model-Based Systems Engineering data. [...] The management of multiple representations of components and subsystems in a system is enabled with ‘variant’ management, allowing the instantiation of a system model, function of the stage of development, and the purpose of the simulation (LMS International, 2012).”

So, LMS Imagine.Lab SysDM is a data management tool focused on behaviour models such as those from LMS Imagine.Lab AMESim or Matlab/Simulink. Additionally, CAE data of other simulation authoring or solving tools such as Nastran input decks are storable. So LMS Imagine.Lab could manage different kinds of simulation data. However, LMS Imagine.Lab only supports the modelling of simulation models created with LMS Imagine.Lab AMESim. This means LMS Imagine.Lab SysDM cannot be used for simulation data and process management in general (for all kinds of models) because the support of the modelling processes is restricted. LMS Imagine.Lab AMESim is an exception because this simulation authoring tool is supported in an SDPM context with LMS Imagine.Lab SysDM. As result, only the executable files generated by external simulation processes are stored in LMS Imagine.Lab SysDM with exception of LMS Imagine.Lab AMESim simulation processes. Additional, LMS Imagine.Lab SysDM is not integrated into a PLM system.



The main focus of LMS Imagine.Lab SysDM is on system-simulation data that is modelled and used at the system level. The system level defined by the system engineering methodology is assumed in this context. LMS noticed in the context of system engineering and the usage of LMS Imagine.Lab SysDM (LMS International, 2011):

”Model-based systems engineering relies on system level models to simulate the overall performance and behaviour of new intelligent products made of complex interactions between mechanical, hydraulic, pneumatic, thermal and electric, electronic phenomena.”

Simulation models that simulate system behaviour, such as a simulation model created in system engineering context, are called behaviour models. Behaviour models represent the behaviour of functions. Multiple behaviour models are mostly dependent upon each other. This is similar to multiple functions that are dependent upon each other. Multiple functions also interact with each other. To represent such interaction of functions, the behaviour models could interact (assuming the required interfaces are available) with each other as an inter-disciplinary simulation. This will be discussed in more detail in section 4.8.1 and 4.8.2. For the modelling of the behaviour models, EBS authoring tools will be used. The EBS authoring tools provide a multi-physical (disciplinary) modelling environment. So, the behaviour model technique will often be used for multi-disciplinary and inter-disciplinary simulation. Nevertheless, the EBS technique will be restricted in the accurateness and detailing of the reality. The modelling of EBS models with high accurateness and detailing will be often uneconomical. Other simulation techniques such as FEM (finite element method), MBS (multi body simulation), SPM, CFD (computational fluid dynamics), etc. provide better technologies for improved accurateness and detailing. In some cases simulation models with improved accurateness and detailing will be required in multi-disciplinary and inter-disciplinary simulation context. As a result, model-based systems engineering will use behaviour models for validation and verification of both products and the underlying product development processes. Thus the combination of systems engineering methodology and system-simulation can be seen to improve the development process, which will be discussed in Section 2.7.

TEAMCENTER can be used for the data management of behaviour models. TEAMCENTER supports the modelling of simulation models with Matlab/Simulink and the system-simulation generation with Matlab/Simulink simulation models. However, the behaviour model's technology is limited to the context of Matlab/Simulink. The architectural concept provides possibilities to add the support of other simulation authoring tools, but this requires API configuration (Siemens Product Lifecycle Management Software Inc., 2012). A detailed data acquisition of the behaviour modelling technology in TEAMCENTER will be documented in Section 4.8. The behaviour model technology is fully integrated into TEAMCENTER and, consequently, into a PLM system.

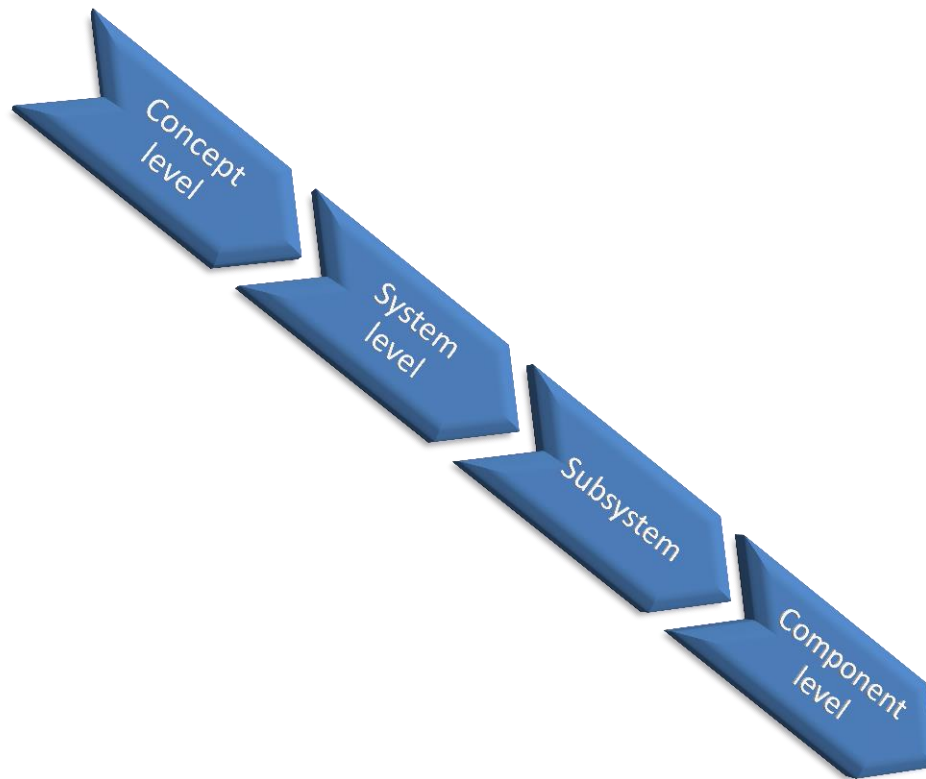
Another system-simulation data management system is the model library that can be added to the simulation middleware ICOS. This system is provided by the international research

centre “Virtual Vehicle” in Graz (Zehetner, Wenpu Lu, Watzenig & Bernasch, 2012). The system is rather a research object than a commercial product and designed solely for internal use by the Virtual Vehicle Company. The system stores EBS library models generated by functional departments. Data management is achieved using metadata. The metadata points to stored library models. This package can then be populated in the system-simulation data management system. The library models and metadata are re-used for the generation of system-simulation models. The system-simulation models are project-oriented (the data is dedicated logically to the project). The software can be seen as a central data store system to manage models and additional simulation artefacts. Only the metadata and the executable simulation files are stored in the presented software. As a result, it is not integrated into a PLM system.

From the discussion above, it can be seen that there is no system-simulation data management software that fully supports a variety of simulation models and system-simulation models. None of them provide an open approach to support the simulation modelling process. This means that no system-simulation data management software can be used to provide SDPM functionalities even though the behaviour model technology of TEAMCENTER provides the integration function to a PLM system. Inevitably, this system-simulation data management system only provides the lowest interaction with other simulation authoring tools.

## 2.7 THE SYSTEM ENGINEERING METHODOLOGY

The purpose of the system engineering or system engineering process is to provide a process that transforms requirements into technical specifications that support a development from the product architecture concept to the detailed virtual product description (Department of Defense, Systems Management College, 2001). The system engineering process summarises and details the needs and requirements from customers into sets of development-level-dependent product and process descriptions. The development-level-dependent sets of requirements and specifications could be compared with the functional description of a system or other kinds of virtual product descriptions. On each level of development, the requirements, specifications, virtual product and process descriptions become more and more detailed. The virtual product and process descriptions can be a functional description of the system, and CAD or CAE models. The system engineering methodology usually progresses through multiple development levels such as the concept level, the system level, then the subsystem and the component level shown in Figure 2-3.



**Figure 2-3 Levels of System Engineering**

At each level there is a level-related product description, such as functional or design data. The product development process is verified at the beginning of each step from one level to the next. This verification is to evaluate the fulfilment of the requirements and specifications of the level-related system solution. The verification procedure is a design synthesis and often follows mini-V model (Berry, 2011; Department of Defense, Systems Management College, 2001) as shown in Figure 4-2. The verification process compares the requirements of a system or subsystem against the achieved functions of the product design. This should be done at each development level. The engineering product design normally starts at the system level with the generation of functions. Then CAD geometrical models will be created during the development process.

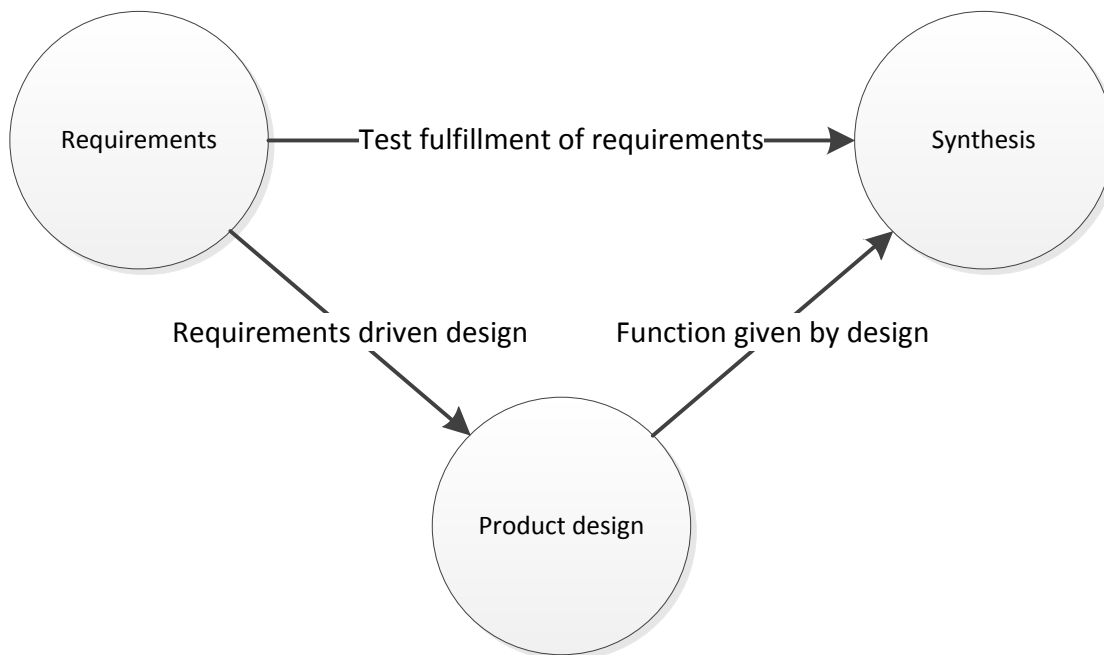


Figure 2-4 Mini V-Model for Verification

The system engineering methodology supports the development of mechatronic systems (Vyatkin, Hanisch, Cheng, & Chia-Han, 2009). The detailing of the product requirements from one development level to the next creates improved input for each subsequent development process step. The INCOSE International Council on System Engineering (2013) comes to the consensus that

“Systems Engineering is an engineering discipline whose responsibility is creating and executing an interdisciplinary process to ensure that the customer and stakeholder's needs are satisfied in a high quality, trustworthy, cost efficient and schedule compliant manner throughout a system's entire lifecycle.”

This definition integrates the viewpoints of the three main standards: MIL-STD-499A “Engineering Management”, EIA Standard IS-632 “System Engineering” and IEEE P1220 “Standard for Application and the management of the Systems Engineering Process.

In addition to the verification of the development process, a validation of the system or product is supported by the system engineering methodology. This verification is used to check the development process against the achievements of an expected system or product development step. Katasonov & Jyväskylä (2008) wrote about the verification:

“Am I building the right product? Checking a work product against higher-level work products or authorities that frame this particular product.”

As for the validation of a product, a system or product prototype has to be developed. That means the validation process can only be done at the end of the system or product design

process. Activities such as testing and the simulation of virtual prototypes belong to the validation process.

The validation process can be based on a component, a subsystem or a system. The checked requirements and specifications of a validation test or validation simulation are dependent contextually to the prototype and the development level (component, subsystem, system). So, a subsystem prototype will mostly be checked against the requirements and specification that should be achieved by the subsystem. The validation of the system and product requirements should be checked at all the development levels. In Figure 2-3 the development levels start with the concept level and go down to the component design development level. At each downward step, a verification of the development process in the mini V-model (see Figure 2-4) will be carried out. Subsequently, the system or product validation process goes upwards beginning with the simulation and testing of component prototypes concluding with the testing of the end system or product. This is called the big V-model and is shown in Figure 2-5.

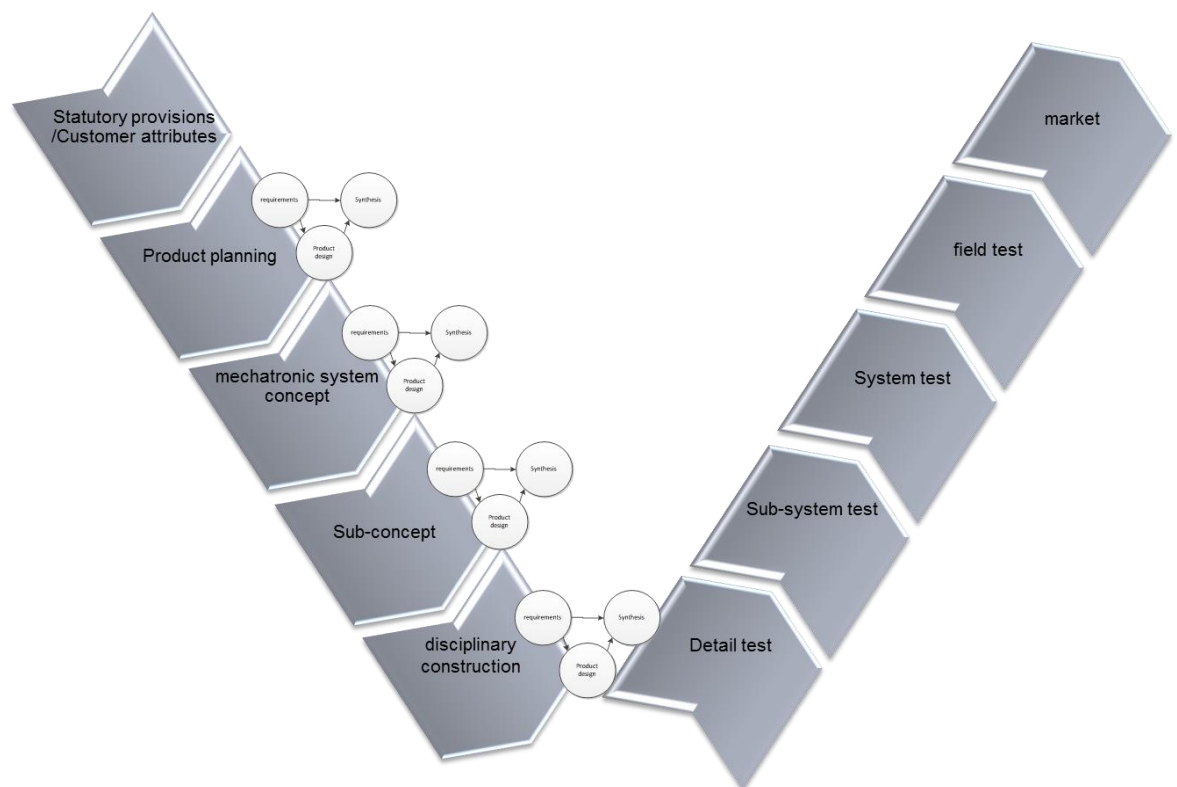


Figure 2-5 Big V-Model

Multi-disciplinary system-simulation plays an important role in the system engineering methodology. It can be used for validation and verification (Zaeh & Baudisch, 2003). Therefore, simulation has to represent a system, subsystem or component (Hui, Liping, Li, & Tifan, 2011). The representation of a system, subsystem or component requires the

simulation of multiple physical disciplines. The simulation models should be kept simple because there is a gap in the access, availability and identification of interacting simulation models. With increasing complexity of the simulation models the requirements to close these gaps will increase also. A common framework ought to close these gaps. Additionally, a common framework should help to plan the simulation models of a system, subsystem or component. For example, a common framework could represent the relationships between data, and support the distributed simulation model generation process with this associated information. So, different simulations could work on a common simulation with sub-ordered simulation models, and be based on system or product descriptions such as functions or CAD models (Zaeh & Baudisch, 2003). However, the most common basis of a behaviour simulation will be functions which will help to keep the simulation model simple as many boundary conditions could be ignored. In several instances, this kind of base will be not ideal although it will reduce the gap of missing data relations, data interaction and data availability. As stated earlier, a common framework should close these gaps and provide better support for such simulations. In addition there is another gap in the support of simulation data lifecycle: The simulation sources, simulation requirements, simulation authoring tools and solving tools changes during the lifecycle. Simulations are achieved for a specific moment in the development process. Often such a moment in the development process will be a milestone. Therefore, a simulation specific view on the simulation data lifecycle is not always required. However, system engineering integrates both verification and validation technology during the product development lifecycle. So, the simulation technologies would be used to check system, subsystem or component behaviours against requirements during each development level. As such, the simulations will be dependent upon the development process and the different development levels and those simulations will be dependent upon the products or functions lifecycle. Additionally, the representation of the required simulation accuracy changes during the development process. To address these changes the simulation authoring tool will require modifications during development processes. Nevertheless, the simulation of the same system, subsystem or component will be applied multiple times with different levels of accuracy throughout different development levels during the product or function development process. This will mean that the simulation underpinning its own lifecycle and will be dependent upon the product or function lifecycle. Again, there will be gaps in the support of lifecycle dependencies and a common framework should remove these gaps and provide improved support.

The new framework to improve the support and management of multi-disciplinary simulation data should provide the data management of the system engineering process with verification and validation simulations. That means that the new framework has to support and manage system-simulation data.

In summary, the system engineering methodology shows that future product development methodologies will require an interaction between different disciplines. This idea of combining different disciplines is similar to that of the PLM system. The TEAMCENTER PLM system of Siemens Product Lifecycle Management Software Inc. (2011) supports that:

“complex. [...] Products require a systems-driven approach to product development that combines systems engineering with an integrated product definition [...]”

Ergo systemengineering theory should be integrated into the product development process and hence in the product lifecycle.

From the above discussion, it can be seen that there are considerable gaps between existing functions of systems and software and those required for achieving full system integration to support and manage multi-disciplinary and inter-disciplinary system simulations in a PLM environment (Zaeh & Baudisch, 2003). The discussion about how to deal with this gap will be examined in Section 2.8.

## 2.8 PROBLEM DISCUSSION

This Section will discuss the gaps between the functional requirements of supporting and managing multi-disciplinary and inter-disciplinary system-simulation in a PLM environment with those of existing commercial products. The gaps are:

- the functions for implementing the universal support of multiple simulation authoring tools in a multi-disciplinary environment
- the functions for supporting the cooperation and interactions between multiple simulations (inter-disciplinary simulation) in a PLM and SDPM environment
- the functions for supporting the different system or product descriptions in multi-disciplinary system-simulation
- the functions for supporting the interactions between different development disciplines involved in a product development process

### 2.8.1 FUNCTIONS FOR THE SUPPORT OF MULTIPLE SIMULATION AUTHORING TOOLS

Most of the current SDPM systems are focused on geometry-based FEM (Finite Element Method) simulation. However, SDPM systems should also be open to other kinds of simulations such as EBS simulation in a controller simulation. Furthermore, it would be ideal if SDPM supports all kinds of CAE software (Dr. Ing. Sippel & Dipl. Ing. Niederauer, 2010, S. 209). The support of different simulation authoring tools includes the ability to execute them and the opening of the simulation models (see Section 2.6 I). Some of the evaluated SDPM



tools meet this requirement. However, only TEAMCENTER for Simulation can offer a high level of transparency when interacting with different simulation authoring tools.

The requirement for SDPM applications exists in the support and management of multi-disciplinary simulation. Multi-disciplinary simulation requires simulation models from multiple simulation authoring tools. But SDPM applications do not currently support and manage multi-disciplinary simulation. Instead of SDPM tools, system-simulation data management tools (see Section 2.6 II) can be used. However, the system-simulation data management solution does not support and manage all kinds of simulation authoring tools (see Section 2.6 II). So, system-simulation data management software solutions are not supporting the generation of system-simulation models interacting with individual simulation models generated by different authoring tools. In addition, the system-simulation data management software solutions preclude the interaction with PDM and PLM.

The previous discussions show that no existing tools can be used to support multiple simulation authoring tools across the required breadth of different domains.

### *2.8.2 FUNCTIONS FOR THE SUPPORT OF THE COOPERATION AND INTERACTION BETWEEN MULTIPLE SIMULATIONS*

In the case of multi-disciplinary-simulation, the different simulation disciplines and simulation tools have to be coupled (Zaeh & Baudisch, 2003). This can be a serial or parallel coupling (for detail see Chapter 6). Parallel coupling is called co-simulation (Geimer, Krüger, & Linsel, 2006). A co-simulation requires multiple simulation models running in parallel. In the case of multi-disciplinary simulation, the data of multiple simulation models has to be managed. Each of the simulation models is generated in a single simulation process that generates a large quantity of files and data (see Section 2.3). The data management of multi-disciplinary simulations includes the management of all these files and data of interacting single simulation models involved in the multi-disciplinary simulation.

Actually, some SDPM tools can support the generation of single simulation models that are CAE vendor-independent. By comparison, no system-simulation data management tools support the generation of single simulation models that are CAE vendor-independent. The CAE vendor independence is important for a common framework because multi-disciplinary simulations mostly require multiple CAE vendor simulation authoring tools. As a result current data and information of an individual simulation and of system-simulations are often not managed by one tool. Importantly, the simulation data has to be managed in a way that the dependency is tracked and traceable (Ulrich, 2011). An individual simulation will be refollowable in the SDPM but not in the system-simulation data management tools. Consequently, traceability from the system-simulation model to the individual simulation model source will fail. Essentially, there is a gap because the cooperation of and interactions between multiple simulations are supported by system-simulation data holding tools, but the



holding of the data in such a system will lose the support of the simulation process and its relationship to those simulation sources. Hence, there will be a gap between the functions of SDPM and PLM in supporting the cooperation and interactions between multiple simulations.

Therefore, in addition to the traceability of individual simulation models, the multi-disciplinary simulation should be traceable to different system or product descriptions.

### *2.8.3 FUNCTIONS IN THE SUPPORT OF DIFFERENT SYSTEM OR PRODUCT DESCRIPTIONS*

Every kind of simulation requires large, varying amounts of different input data and information. The simulation result quality is dependent upon the quality of this data and information. A high quality of the input data and information will result in superior simulation results, thereby making the simulation more realistic.

Not only mono-disciplinary simulations, but also multi-disciplinary simulations require input data and information. In a system engineering methodology, a multi-disciplinary simulation is used for validation or verification of systems or products. The systems or products can be described in different ways such as CAD models or function descriptions (see Section 2.7). Multi-disciplinary simulations are based on these system or product descriptions (Zaeh & Baudisch, 2003).

Changes of simulation sources during the development process could considerably affect simulation results. Therefore, simulation source changes have to be reflected by the simulation responses. This requires traceability of the simulation data and information to its associated input data and information. Traceability is also required in the case of reviewing a frozen simulation. A frozen simulation means that all data will be frozen and unchangeable. Frozen data will be required for example to support on-going processes in production and restrict uncontrolled parallel data changes. Freezing will keep all data and information at a fixed stage, thereby making a simulation reviewable, checkable and traceable back from the simulation model to those simulation sources.

Most current SDPM systems are focused on geometry-based FEM simulations, i.e. they are focused on CAD model sources. System-simulation data management tools are focused on managing simulation source metadata information rather than on simulation source data and information. LMS Imagine.Lab supports an additional interface to generate the metadata imported from sysml-files (OML SysML, 2012). However, there are no tools that are supported by several system or product descriptions in the context of multi-disciplinary simulation data management. SDPM can help make individual simulation unique and traceable to source data. Source data holding systems are normally PDM and PLM systems. However, no technologies are available to make multi-disciplinary system-simulation unique and traceable to that source.

#### *2.8.4 FUNCTIONS FOR THE SUPPORT AND INTERACTION BETWEEN DIFFERENT DEVELOPMENT DISCIPLINES*

As mentioned above, the generation of simulation requires high amounts of input and information. Such data and information are generated during the development process and different development tasks. These tasks can be completed by different departments and experts.

There are different kinds of product description information and data. Typical examples are functional descriptions of systems or products and concept descriptions such as sketches or concept models, detailed by mechanical or electrical CAD authoring tools. Correspondingly, the formats of these descriptions are different. So, the production process of a system or product can influence the lifecycle of a system or product's approach. This is the reason why the production process should be both considered, and included, in the simulation. Since this data, information and knowledge are generated by different departments, the corresponding simulation should have access to the data and information of those departments.

In Section 2.7 it was ascertained that validation and verification affect the development process. Simulations could be used for those validation and verification tasks. To make verification or validation simulations effective during the development process, the information and data flow of the simulation to other departments should be properly organised. This could be achieved manually or automatically. A data management system should assist the data and information during a data flow process, for example in data and information push, pull, input and export. As such, the data and information flow are important for system simulation. So, other simulation systems in the development process should be able to access the data and information of the simulation. In this area of system functions, there is an issue. That is, SDPM, PLM, PDM and system-simulation data holding approaches do not support and manage the interactions between different development disciplines in the case of multi-disciplinary system-simulation.

The idea of a cooperative development and department-overlapping database is discussed in Section 2.5. PLM systems try to provide the functionality for cooperative data management between different development departments. But few commercial PLM systems include SDPM systems. The evaluation identified that TEAMCENTER is a leading PLM system. However, a PLM system supporting multi-disciplinary simulations was not identified. Similarly, system-simulation data management tools are independent tools and do not have the functionality to interact with PLM tools.

So a new and improved framework for supporting and managing multi-disciplinary simulation data is required. This will be discussed in Section 2.9.

## 2.9 PROPOSITION OF A NEW FRAMEWORK

As discussed in Section 2.8, there are gaps between the functions of the system for fully supporting and managing the data and information, and those of the existing systems in a PLM environment. These disparities result in a reduced efficient product development process and lowerd quality developed systems or product. Therefore, it is important to propose and develop a new framework to minimise discrepancies. The new framework can be used to support and manage multi-disciplinary simulation data and to improve development processes and systems or products. Additionally, such a new framework could be used to support development methodologies, for example, the system engineering methodology discussed in Section 2.7 could be improved to be a system-simulation based system engineering methodology (see Section 9.4). This means that the validation tasks in the system engineering methodology could be executed by simulations throughout the development process and at each engineering level. For the support of the system engineering methodology, it will be necessary to integrate the support and management of multi-disciplinary simulation integrated in a PLM environment (see Section 2.8.4).

As mentioned in Section 1.1, the full development of this new framework is impossible in the given time and with the given budget of the research project. Multiple resources such as programming experts, IT specialists and software architects are necessary for the full development. But the costs of these resources are well beyond the project budget. However it is feasible to conceptualise and design the new framework. It is expected that the conceptualisation and design of the new framework will provide enough information and knowledge to find the answers to the research questions presented in Section 1.2 and hence to achieve the corresponding research objectives.

## 3 RESEARCH METHODS

### 3.1 INFORMED TOPICS

An improvement of multi-disciplinary simulation could be achieved by the management and support of multi-disciplinary simulation models. This will include the support of overlapping process tasks such as the pre-, post- and solve-processes (Bäuerle-Mahler, 2011, S. 6) (see 5.1 Simulation Process Basics). Existing technologies of the individual simulation process tasks and existing data management solutions could be re-used. Nevertheless, there will still be a lack in the support of overlapping cooperations in the case of multiple simulations and their tasks. A complex system simulation will require multiple experts and multiple software applications to achieve multiple simulation process tasks. The idea is to conceptualise a new framework to support and manage multi-disciplinary system-simulation in a PLM environment.

The research of conceptualizing such a new framework could be achieved based on the post-positivist research philosophy.

#### 3.1.1 ONTOLOGICAL VIEWPOINT

Davison (1998) discusses:

“Two major research philosophies have been identified in the Western tradition of science, namely positivist [...] and interpretive. [...] Positivists believe that reality is stable and can be observed and described from an objective viewpoint. [...] Interpretivists contend that only through the subjective interpretation of and intervention in reality can that reality be fully understood.”

In the discussion, the positivist likes to analyse phenomena which should be isolated and the observations should be repeatable:

“Positivist researchers believe that they can reach a full understanding based on experiment and observation. Concepts and knowledge are held to be the product of straightforward experience, interpreted through rational deduction.” (Davidson, 1998)

In the research project, the new framework should solve the phenomena of supporting and managing multi-disciplinary simulation. The observation will be based on the analysis of multi-disciplinary simulation case examples. Davison(1998) also discusses:

“This often involves manipulation of reality with variations in only a single independent variable so as to identify regularities in, and to form relationships between, some of the constituent elements. [...] Predictions can be made on the basis of the previously observed and explained realities and their interrelationships.”

Variations of variable (variable-studies) will not be realised. Nevertheless, analysis of the case examples will identify regularities of multi-disciplinary simulation processes.

Ryan(2006) discusses the two main positivistic philosophy views. The post-positivistic philosophy view will be discussed: "Post-positivist research principles emphasise meaning and the creation of new knowledge". The research of the post-positivist will create "[...] movements that aspire to change the world". The new framework should change the world of multi-disciplinary-simulation using new knowledge about multi-disciplinary simulation data management. Ryan (2006) continues elaborating on post-positivistic research: "Research is broad rather than specialised – lots of different things qualify as research". In order to conceptualise the new framework, multiple influences on the multi-disciplinary simulation process coming from different disciplines have to be analysed. Theory and practical case examples will be combined in the research project to conceptualise a new framework:

"Theory and practice cannot be kept separate. We cannot afford to ignore theory for the sake of 'just the facts'. [...] The researcher's motivations for and commitment to research are central and crucial to the enterprise" (Ryan, 2006 ).

The research project will use the ontological perspective of a post-positivist approach. The use of the post-positivistic viewpoint will influence the ontological base. This will be discussed in Section 3.1.2.

### 3.1.2 ONTOLOGICAL BASE

The ontological base can be described by research types such as discussed in Kothari (1990). The main types are:

- Analytical vs. Descriptive
- Applied vs. Fundamental
- Quantitative vs. Qualitative
- Conceptual vs. Empirical

#### I. Analytical / Descriptive Research

In (Ryan, 2006) it is stated that the post-positivist approach concentrates on learning, not on testing. This means that post-positivist researchers mostly use available facts and information to create new knowledge:

"Post-positivist researchers believe that positivist research methods predominantly mirror the representational ideology of the positivist researchers. Where the positivist researcher might strive to discover objectively the truth hidden in the subject's mind [...]" (Ryan, 2006).

In contrast to the analytic research, a descriptive research could be done. Descriptive research will include surveys and different kinds of fact-finding enquiries. The user of this research type has no control over the variables. He/she can report what has happened. An analytic researcher will use available facts or information. He/she analyses these facts and information to make a critical evaluation (Kothari, 1990).

The research project will re-use multi-disciplinary simulation case examples from industry. The existing data will be analysed. The required information is available as industry or research-oriented reports or self-made project work. The research project will use the analytic research.

## II. Applied/ Fundamental Research

Principally, the applied research type will be used to find solutions for immediate problems facing a society, industrial or business organization. The applied research type will be used, for example, for marketing research. In contrast to the applied research, the fundamental research type takes care about generalizations and the formulation of theories, specifically for natural phenomena or pure mathematics (Kothari, 1990).

The analysis of the multi-disciplinary simulation case examples will be based on a case example description. Sources of case example descriptions will be of the applied research type. Therefore, I will analyse documents that are based on the applied research type to create a generalised framework to support and manage multi-disciplinary simulation data. This kind of work is a fundamental research type.

## III. Quantitative/Qualitative Research

This research project will not carry out validation test work because the new framework will not be fully developed. It will also not be possible to measure the relationships between the different disciplines taking part in multi-disciplinary simulation. Kothari (1990) states that the quantitative research type will be “based on measurement of quantity or amount. It is applicable to phenomena that can be expressed in terms of quantity.” So, the quantitative research is not ideal for the research project.

On the other hand, Kothari(1990) discusses the qualitative research type: “Qualitative research, [...] is concerned with qualitative phenomenon [...]”. The multi-disciplinary case examples will be based on reports and documents that discuss the workflow and tools of multi-disciplinary case examples. These case example sources and descriptions will include more qualitative than quantitative information which means that the qualitative research is more suitable to this project. Kothari (1990) also discusses the qualitative research type:

“Qualitative research is especially important in the behavioural sciences where the aim is to discover the underlying motives of human behaviour.” So, the research project focuses on qualitative research.

#### IV. Conceptual/ Empirical Research

The conceptual research type seems to be closed to the conceptualisation of the new framework. According to Kothari (1990), the conceptual research type will be “[...] related to some abstract idea(s) or theory. It is generally used by philosophers and thinkers to develop new concepts or to reinterpret existing ones.” At first sight, this perspective seems to be similar to the planned research project. However, the new framework is based on existing case examples with existing solutions. There will be individual solutions available for specific technical problems.

Different case examples will be used to identify general consistencies of a general concept. These will be closer to the empirical research type. The empirical research type is described by Kothari (1990) as relying “[...] on experience or observation alone, often without due regard for system and theory” and “it is data-based research, coming up with conclusions which are capable of being verified by observation or experiment.” The case examples will therefore provide an observation and experimental base which will be used to identify and verify a general concept of the new framework. Consequently, the empirical research type will be used in the research project.

The ontological base will help to keep focused on a clear research. However, to keep the clarity of the research work within boundaries, an ontological structure should help. This ontological structure will be discussed in Section 3.1.3.

##### *3.1.3 ONTOLOGICAL STRUCTURING OF THE RESEARCH PROJECT*

The research project is influenced by a variety of research themes. To structure the research, a solution will not be found through a classical ontological approach. Fernandez-Lopez & Corcho (2004) wrote: “Until the mid-1990s”, the ontology development process “was an art rather than an engineering activity.” Since 1990, designers are working on an ontological development process that will also be useful for engineering:

“In 1996, the first workshop on Ontological Engineering was held in conjunction with the 12<sup>th</sup> European Conference on Artificial Intelligence. Its goal was to explore a suite of principles, design decisions, and rules of good practice from which other ontology designers could profit.” Fernandez-Lopez & Corcho (2004)

One improvement of ontological engineering is a more precise definition of special terms. Terms like methodology, method, technique, process and activity are used in literature indiscriminately (Hoog, 1998). The Institute of Electrical and Electronics Engineers (IEEE) (1990) defines these terms succinctly:

- The methodology is “a comprehensive, integrated series of techniques or methods creating a general systems theory of how a class of thought-intensive work ought to be performed.”
- The method is a set of “orderly process(es) or procedure(es) used in the engineering of a product or performing a service.”
- A technique is “a technical and managerial procedure used to achieve a given objective.”

Ontological engineering could help to link the information and results of different activities and tasks.

The research project used four processes to identify and detail gaps/disparities which are discussed in Section 2.8. These gaps will be re-used as processes which will be described as “what to do’s” instead of research questions according to a view of engineering research.

- Process 1: Conceptualisation of a useful supporting framework interacting with multiple simulation authoring tools.
- Process 2: Conceptualisation of a useful supporting framework for the cooperation of simulation models generated by multiple simulation authoring tools.
- Process 3: Conceptualisation of a useful supporting framework for the interaction of multiple system and/or product-describing structures.
- Process 4: Conceptualisation of a useful supporting framework for relating data and information of different development-disciplines involved in a product development process.

These processes will now be discussed in more detail from an ontological engineering perspective.

## I. Conceptualisation of a Useful Supporting Framework Interacting with Multiple Simulation Authoring Tools

This process starts with the sampling of case examples of multi-disciplinary simulations. The simulation models and the sources should be managed by a PLM system. The PLM system should provide a capability to manage data of multiple simulation authoring tools. These case examples will be analysed from the viewpoint of the interaction between the data storing system and the simulation models generated by multiple simulation authoring tools. Therefore, the previous PLM system has to be evaluated from the simulation data



management point of view. Subsequently, the analysis results can be re-used for the conceptualisation of the new framework.

In Process 1, there are four activities, as follows:

- Activity 1.1: Sampling of multi-disciplinary simulation case examples
- Activity 1.2: Data acquisition of the simulation data management possibilities of the PLM system
- Activity 1.3: Analysing the case examples simulation authoring tools interaction.
- Activity 1.4: Conceptualisation of the new framework

## II. Conceptualisation of a Useful Supporting Framework for the Cooperation of Simulation Models Generated by Multiple Simulation Authoring Tools

This process starts with the sampling of case examples of multi-disciplinary simulations. The simulation models and sources should be managed by a PLM system. The PLM system should provide a capability to manage data of multiple simulation authoring tools. These case examples will be analysed from the viewpoint of multi-disciplinary simulation process and of data export and import. Therefore, the previous PLM system has to be evaluated from the viewpoint of simulation data management. Afterwards, the results obtained from the analysis can be re-used for the conceptualisation of the new framework.

The sampling of the multi-disciplinary simulation case examples and the evaluation of the simulation data management provided by the PLM system can be overtaken from Process 1.

In Process 2, there are five activities, as follows:

- First Activity of Process 2 using Activity 1.1: Sampling multi-disciplinary simulation case examples
- Second Activity of Process 2 using Activity 1.2: Data acquisition of the simulation data management possibilities of the PLM system
- Third Activity of Process 2 completed in Activity 2.1: Analysing the multi-disciplinary simulation process of the case examples.
- Fourth Activity of Process 2 completed in Activity 2.2: Analysing the ex- and import during the multi-disciplinary simulation process of the case examples.
- Fifth Activity of Process 2 using Activity 1.4: Conceptualisation of the new framework

### III. Conceptualisation of a Useful Supporting Framework for the Interaction of Multiple System and/or Product-describing Structures

This process starts with the sampling of case examples of multi-disciplinary simulations. The simulation models and the sources should be managed by a PLM system. These case examples will be analysed from the viewpoint of the different system or product descriptions and the interaction between the system and product descriptions. The analysis results can be re-used for the conceptualisation of the new framework.

The sampling of the multi-disciplinary simulation case examples can be taken from Process 1.

In Process 3, there are three activities, as follows:

- First Activity of Process 3 using Activity 1.1: Sampling multi-disciplinary simulation case examples
- Second Activity of Process 3 completed in Activity 3.1: Analysing the system and product descriptions of the case examples.
- Third Activity of Process 3 using Activity 1.4: Conceptualisation of the new framework

Process 3 could re-use the analysed results of Processes 1 and 2 as input for the Activity 3.1. Therefore, the analysis results of the simulation authoring tool interaction with the SDPM and the multi-disciplinary simulation process can be re-used for the analysis of the system and product descriptions.

### IV. Conceptualisation of a Useful Supporting Framework to Relate Data and Information of Different Development-Disciplines involved in a Product Development Process

This process starts with the sampling of case examples of multi-disciplinary simulations. The simulation models and the sources should be managed by a PLM system. These case examples will be analysed by relating the different data and information of the systems or products with the simulation. Therefore, the previous PLM system has to be evaluated using simulation data management. Subsequently, the analysis results can be re-used for the conceptualisation of the new framework.

The sampling of the multi-disciplinary simulation case examples and the evaluation of the simulation data management provided by the PLM system can be taken from Process 1.

In Process 4, there are four activities as follows:

- First Activity of Process 4 using Activity 1.1: Sampling multi-disciplinary simulation case examples
- Second Activity of Process 4 using Activity 1.2: Data acquisition of the simulation data management possibilities of the PLM system
- Third Activity of Process 4 completed in Activity 4.1: Analysing the data and information relations of the case examples.
- Fourth Activity of Process 4 using Activity 1.4: Conceptualisation of the new framework

Activity 1.4, the last Activity in processes 1 to 4, could re-use the previous activity results of each process as input. Therefore, the analysis of the simulation authoring tool interaction with the SDPM, the multi-disciplinary simulation process and the analysis of the system and product descriptions can be re-used to analyse information relationships.

### 3.2 USED METHODOLOGY AND METHODS

The research project is based on case studies. According to Davison (1998), the methodology and method of case studies will “[...] involve an attempt to describe relationships that exist in reality[...].” Therefore, data has to be collected and analytical techniques should be employed. Relationships of data will be one of the main focuses of the research project.

Case studies are discussed by Benbasat, Goldstein, & Mead (1987) and Davison (1998). The researcher can ask “how” and “why” questions, so as to understand the nature and complexity of the processes taking place. Such questions are answered by analysing case examples. Multiple case examples will be sampled, as discussed in Benbasat, Goldstein, & Mead (1987):

“A case study examines a phenomenon in its natural setting, employing multiple methods of data collection to gather information from one or a few entities [...].”

The case examples are not controlled or manipulated by the analysis as discussed in Davison (1998): “[...] [I]f there is a need for control or manipulation of variables, then the case study would not be appropriate.”

So, the case study methodology and method will be useful for all activities. There are three different types of sampling in the ontological structuring of the research project. The first one is the sampling of the case examples used in Activity 1.1. The second one is the sampling of data via about the simulation data management in Activity 2. 1 and the third sampling is the re-use of analysis data, and of previous analysis activities and tasks such as those in the Processes 3 and 4.

The sampling was discussed in Davison (1998): “[C]ase studies require multiple data collection methods, whose results hopefully converge, in order to establish construct

validity.” Yin (1984) explains that sampling can be done with documents (written, printed or electronic) as well as records and charts about companies and their operations or their previous use of technology. The research project will use papers, project documents and personal knowledge. The documents and personal knowledge will largely be based on the author’s own previous business projects, i.e., the sampling technology of the case examples of Activity 1.1.

The methodology and method of protocol studies are similar to case studies. Protocol studies are discussed in Green, Kennedy & McGown (2002):

“Engineering design research has often placed credence in data taken from observation of designers [...] Protocols involved observation of designers at work.”

This fits this project, but they also explained: “Almost all of these studies are based on what we might call ‘experimental data’[...]” However, this will not be the case in this thesis because the sampled protocols do not describe the experimental process; they describe simulation processes or the use of data management. Dwarakanath & Wallace(1995) and Green, Kennedy & McGown (2002) explained:

“[R]ecognise the shortcomings of such experiments in saying that it is ‘less representative for analysis of how design actually takes place in practice’. Acknowledgement of this caveat helps to bolster the credibility of their protocol studies, and their claim that a laboratory environment ‘usefully restricts the influences on the design processes”

According to Green, Kennedy & McGown (2002):

“The resulting range of paper reveals that even though they might be based on evidence gained in ‘controlled laboratory environments’, there are still many ways of interpreting the results.”

If the simulation is considered as a virtual art “controlled laboratory environments” then the methodology and method of protocol studies describes the work in the research project. The analysis in Activities 1.3 and 1.4 plus the conceptualisation of all activities will re-use the protocols of the case examples and their activities. In this case, protocol study means to analyse experimental protocols. Simulations can be seen as a virtual experiment. So protocol studies can be considered as an analysis of simulation protocols. The sampling of Activity 2 will re-use protocols of data management systems, which cannot be viewed as precise experimental documentations but instead as a manual to make data management.

In order to generate new knowledge in multi-disciplinary simulation data management, the case study and protocol study methodology and methods will be used. With these methodologies and methods, existing data management technologies will be compared with multi-disciplinary simulation case examples. Therefore, data sampling of case examples as well as data management technologies are required. The data sampling will be discussed in Section 3.2.1.

### 3.2.1 DATA SAMPLING FOR THE DATA ACQUISITION OF DATA MANAGEMENT TECHNOLOGIES

The sampling of data on simulation data management technologies is complex, due to the different software tools and systems that could be in use. A reduction to one relevant data managing system should help keep the research project focused. An evaluation of different systems will help to identify the relevant data management system. Based on this data management system, the possibilities of simulation and multi-disciplinary simulation data management should be achieved. These will present the actual possibilities of simulation and multi-disciplinary simulation data management.

The acquired data will provide the basis to understand the data management of the multi-disciplinary simulation case examples. Additionally, such data provides the possibility to compare those case examples with the existing technology of simulation and multi-disciplinary simulation data management.

This kind of work will divide Activity 1.2 into four tasks:

- Activity 1.2: Data acquisition of simulation data management possibilities of the PLM system
  - Task 1.2.1: Data sampling of existing data management systems
  - Task 1.2.2: Evaluation of data management systems
  - Task 1.2.3: Data sampling of simulation data management of a selected data management system
  - Task 1.2.4: Data acquisition of simulation data management technology

The first two tasks (1.2.1 and 1.2.2) will be included in the theoretical discussion of the research project in Chapter 2. Tasks 1.2.3 and 1.2.4 will be discussed in Section 4. The data sampling of Task 1.2.3 will be the easier option. This data sampling is focused on data that describes the handling of a data management tool. Documents such as product manuals and guides, online help, workshops, training documents and marketing papers will form the basis of this data acquisition, and as such, will be based on an evaluation of the data management systems.

In contrast to this, the data sampling (Task 1.2.1) will be much more complex. Different kinds of data management tools are on the market and can have different effects on users. So, a theoretical discussion on data management and its effects according to the different approaches and the user requirements should achieve a better understanding. Therefore, documents such as whitepapers, marketing documents or other product descriptions and data management survey documents and papers about data management will be used. This will provide the basis for the evaluation of data management systems.

Data acquisition of simulation data management will help review and analyse the used data management of case examples. This will be carried out in the case study of the research project. The case study will be discussed in the following section.

### *3.2.2 DATA SAMPLING FOR THE ANALYSIS OF THE CASE EXAMPLES AND CASE STUDY*

In these case studies, the data and processes of the case examples will be in a data-managed mode. The acquisition of the data management tool in Activity 1.2 will help to understand the case examples. The case studies will be analysed to figure out:-

- the support technology of simulation authoring tools interaction (Activity 1.3) and
- the support technology of multi-disciplinary simulation process (Activity 2.1) and
- the support technology of data export and import during the multi-disciplinary simulation process (Activity 2.2),
- the support of system and product descriptions (Activity 3.1) and
- the support of data and information relationships (Activity 4.1).

The sampling of the case examples plays an important role. They define the basis for analysis and usability of the new framework. In order to define an optimistic basis for a new framework, the case examples should provide a robust state of technology for future technologies. The examples should have authenticity and represent the actual interest of the German market for system-simulation data management. The intention is to identify case examples that are relevant for a market with growing potential. My work as CAE consultant at Siemens Industry Software GmbH & Co.KG was helpful in identifying case examples. The following four projects have been identified to meet both requirements and boundaries:

- Customer: Semi-finished goods producer (unpublished customer name), South-Germany  
Project: Benchmark of simulation tools. This benchmark should identify an ideal simulation tool. An improved development process of extrusion section production tools was achievable with Siemens simulation products.
- Customer: Engineering company SCHMIDT Gesellschaft für Werkzeug- und Formentechnik mbH; Nürnberg  
Project: Creation of a moulding tool system-simulation vision for moulding tool optimization.
- Customer: Automotive Company (unpublished customer name), South-Germany  
Project: The one-vendor-benchmark project. In this project, a multi-disciplinary and mechatronic system-simulation of a car luggage door system had to be achieved.
- Customer: Automotive Company (unpublished customer name), South-Germany  
Project: A research project called "Interdisciplinary Model-based Development Process". This research project included a multi-disciplinary and mechatronic system-simulation of a car windows lifter system.

These four case examples will be presented and discussed in Section 5. Sub-ordered sections will also include analysis of the case examples. More case-example-overlapping analysis will be carried out in Section 6. This analysis will provide the basis for the conceptualisation of the new framework to support and manage multi-disciplinary simulation data. The conceptualisation will be discussed in Section 3.2.3.

### *3.2.3 CONCEPTUALISATION OF THE NEW FRAMEWORK*

Section 6 will conceptualise the new framework which will be based on case example analysis. However, the conceptualisation cannot be achieved in one step. Multiple improvements and architectural instance-concepts need to be discussed and presented. The final architectural approach should summarise the different architectural instances and improvements.

The new framework could achieve different benefits and can be validated based on these benefits. So, the benefits of the new framework should be tested in relation to the case examples. The benefits and the validation will be discussed in Section 6.6.

Unfortunately, the new framework could not be built up, generated and completed during the research project because of lack of time and funding. So testing and validation of the realistic framework will not be possible. Instead, a validation of benefits achieved by the new framework should be carried out which will be achievable through impact analysis. The impact analysis will be completed by analysing the improvements of case examples data management using the new framework.

This kind of work will divide Task 1.1.4 into three tasks:

- Activity 1.4: Conceptualisation of the new framework
  - Task 1.4.1: Architectural new framework approach
  - Task 1.4.2: Detailed new framework approach
  - Task 1.4.3: Impact analysis of the case example data management with the new framework concept



## 4 DATA ACQUISITION OF THE SIMULATION SUPPORTING SYSTEMS AND TECHNOLOGIES IN TEAMCENTER

This section describes the acquisition of simulation data management technologies. Multiple software tools, systems or applications can be used for the acquisition. However, as discussed in previous sections, there is no ideal software for such acquisition.

The requirement for a data management tool that supports multi-disciplinary simulation data embedded in a PLM system was determined in Chapter 2. The evaluation and analysis determined the PLM system TEAMCENTER as a leading technology which includes SDPM described in Sections 2.6 through 2.9. The existing SDPM of TEAMCENTER Univied Architecture 9.1 appears to be the best data collection system for the proposition of the new framework.

The data acquisition will be analysed to obtain detailed knowledge about SDPM technologies employed in TEAMCENTER Univied Architecture 9.1.

First, an overview of the data models in TEAMCENTER will be generated, which will form the basis of those models, i.e. the fundamental knowledge that is necessary to understand the on-going simulation data models with the specific CAE ITEMS and CAE Relations.

### 4.1 DATA ACQUISITION OF THE BASIC DATA MODEL OF TEAMCENTER

The data model of TEAMCENTER is based on items.

“Items, item revisions, documents, parts, and designs are business objects, and as such, are fundamental data objects used to manage information in TEAMCENTER. Items are structures that are generally used to represent a product, part, or component. Item revisions are data objects used to manage changes (revisions) to items. The item type is the parent type for many objects, including document, part, and design types” (Siemens Product Lifecycle Management Software Inc., 2012)

The ITEM does not change throughout the lifecycle of a product, part or component, although changes to products, parts or components can occur, based on the concept that a product, part or component is optimised to fulfil specific requirements. During the engineering lifecycle, changes are realised, for example through optimization of stress or deformation and creating ideal interfaces between products, parts or components, by designing screw holes or other geometrical features. In the released status, it could be necessary to change suppliers of the product, parts or components or to improve them. In summary, the description of products, parts or components will change throughout the lifecycle. In most cases, changes of those products, parts or components need documenting and such documentation will prove to be significant since multiple users will often be working with the same product, part or



component. Used data will need to be checked and tracked for authenticity and release status. Therefore, TEAMCENTER provides item revisions. Multiple item revisions will be structured under each item. The changed product, part or component (after the competition of the change, which could have duration of hours or days) is managed under a new item revision.

“An item in TEAMCENTER is a structure of related objects. The basic structure of any item consists of the following minimum objects:

Item:

Collects data that is globally applicable to all revisions of the item.

Item Master (Form):

A form object that is often used to extend the stored property data for an item to include data unique to the customer.

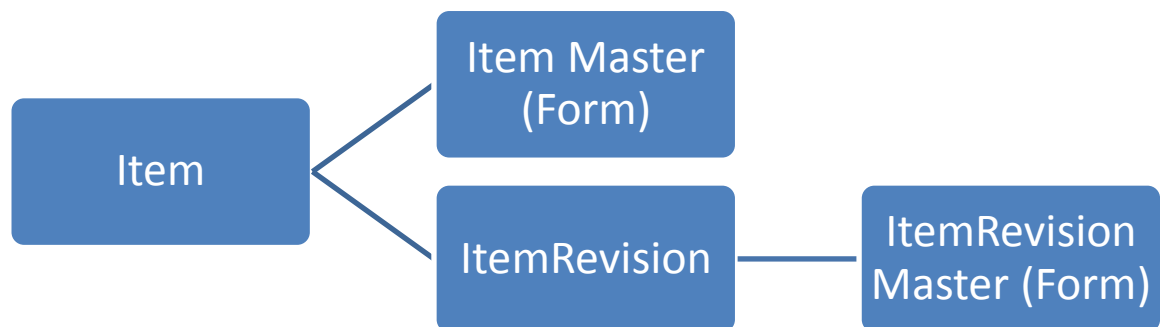
Item Revision:

Collects data that is applicable to a single revision of the item.

ItemRevision Master (Form):

“A form object that is often used to extend the stored property data for an item revision to include data unique to the customer.” (Siemens Product Lifecycle Management Software Inc., 2012)

The Figure 4-1 monitors the data model.



**Figure 4-1 Basic Minimum Structure**

“Some applications provide specific items. For example, the TEAMCENTER simulation process management applications use CAEModel items, CAEGeometry items, and others.

[...]There are typically many pieces of information that describe or are related to an item or item revision. TEAMCENTER uses relations to define the correlation between data objects and items or item revisions.

[...]

Many item or item revision relations are automatically defined when you create or add certain objects to an item or item revision structure. For example, when you add a new item revision to an item, the new item revision is automatically defined as a revision relation.” (Siemens Product Lifecycle Management Software Inc., 2012)

In the case of strings, additional files, data or information are added to an item or item revision. In the case of global files, data or information, the dataset is linked to the item. In the case of files, data or information that could change during the lifecycle is linked to the item revision.

“The Dataset object represents an actual data file on the operating system or in TEAMCENTER. Datasets are typically authored content of some sort, such as Microsoft Office files or CAD data files.” (Siemens Product Lifecycle Management Software Inc., 2012)

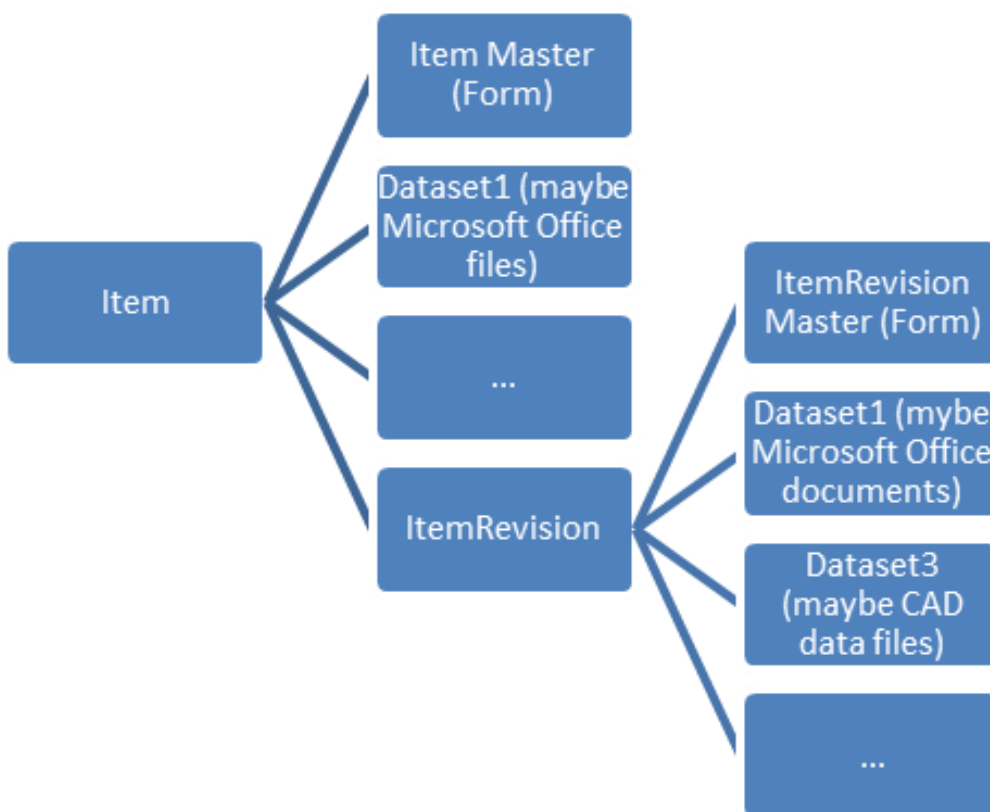


Figure 4-2 Item revision - Dataset Structure

The hierarchical structure of assemblies is realized in TEAMCENTER through a hierarchical structure of CAD ITEM Revisions as shown in Figure 4-2. In the case of an assembly, a data structure including the subordinated CAD ITEM Revision is stored.

“When you add a component to an assembly, you are creating an occurrence of that item or item revision in the assembly, which is stored on the BOM view revision. This occurrence is displayed as a BOM line. A BOM view revision is a single-level structure that contains occurrences of its immediate children. A multilevel structure is built up from many single line BOM view revisions.” (Siemens Product Lifecycle Management Software Inc., 2012)

These technologies are also used for the simulation data model in TEAMCENTER. Special ITEM types are available to manage special files, data and information. The simulation data model in TEAMCENTER is discussed in Sections 4.2.

## 4.2 DATA ACQUISITION OF THE SIMULATION DATA MODEL IN TEAMCENTER

Simulation files, data and information should be easily identified as simulation objects in the PLM system. Therefore, the simulation files, data and information are items derived from the basic data model of TEAMCENTER.

“CAE items are the fundamental workspace objects in TEAMCENTER used to manage CAE information. [...] CAE items are generally used to maintain the CAE representation of a product, part, or component. They also maintain the definition of the analysis performed on these items, and the results of the analysis.” (Siemens Product Lifecycle Management Software Inc, 2012)

The simulation data model of TEAMCENTER uses four items. Each of these items stores data on another level in the simulation process. The simulation process is structured as pre-process, solve-process and post-process. Pre-process prepares the data for the solve run. The solve run calculates a mathematical solution of the problem and the post-process visualises, interprets and analyses the simulation results. Data generation is dissimilar to the simulation process. TEAMCENTER supports four phases of data-generation in a standard configuration:

1. The first phase is the generation of a geometrical model in the pre-processing phase: This geometrical model or geometrical description will prepare data for the abstraction made in the next step. For example, the generation of a given CAD model could include the deletion of simulation-unimportant holes or blends, such as holes and blends with a small radius. Holes and blends with a small radius have no or minimal effect on the simulation result. Therefore it makes no sense for the simulation result to include them, plus deletion of these unimportant holes and blends reduces the abstraction work and simulation solve time. These files, data and information are managed by the CAEGeometry item and item revision.
2. The second phase is the abstraction to a mathematical geometrical description in the pre-processing phase: Based on the (for abstraction) prepared geometrical data, the data is abstracted to a mathematical geometrical description format. Dependent on a simulation discipline such as FEM, MBS or others, the geometrical description is based on the mathematical handling used by the simulation discipline. For example, in the case of FEM, the mathematical handling is based on finite elements and finite elements are geometrically described by finite element meshes, i.e. the idealised geometrical data is abstracted to the description of the geometry by finite element meshes. These files, data and information are managed by the CAEModel item and item revision.

3. The third phase is the modelling of load cases, boundary conditions and the adding of solution parameters in the pre-processing phase: The abstract description of the geometry is not the description of the simulation. Load cases and boundary conditions are missing. Load cases such as forces on nodes (which are the element edge points) or boundary conditions such as freedom degrees on nodes are required to fulfil the abstract description of the simulation. For example, the mesh of a beam bounded on a wall and on the other side a load force represents the force influence of a moving mass. Additionally, parameters required for the simulation solve run are required. Based on the geometry abstraction data, the load cases and boundary conditions and the solve parameters data are added. These files, data and information are managed by the CAEAnalysis item and item revision.
4. The fourth phase is the results of data generated during the solve- and post-processing phases. Based on the pre-processing phase, the simulation is solvable. A solve run of a simulation generates result files. The data generated by the solver is linked to the solve run phase, for example, the stored files of the calculated stress or deformation in the solve run. The on-going work on the result files to visualise, interpret or analyse the simulation results are linked to the post-processing phase. An example of this could be a diagram of the stress or deformation, and appraisal documentation produced by the analyst. These files, data and information are managed by the CAEResult item and item revision.

Other solutions to support simulation phases are possible and additional kinds of supporting simulation phases are configurable with TEAMCENTER. The Table 4-1 gives an overview of the different simulation ITEM types.

Table 4-1 CAE Item Classes (Siemens Product Lifecycle Management Software Inc, 2012)

CAEITEM – Classes	Purpose
CAEGeometry	<p>“Contains the idealised geometry used to generate model data for performing engineering analysis.</p> <p>The contents of the CAEGeometry item can be created from the geometry contained in a non-CAE item revision. It can often have simplifications such as feature suppressions to the original product geometry to facilitate the analysis process. CAE geometry acts as a source representation upon which a CAE model is built.” (Siemens Product Lifecycle Management Software Inc, 2012)</p>
CAEModel	<p>“Defines the model used for performing engineering analysis. In the case of finite element analysis, this model is likely to be in the form of mesh data. In general, any type of model is supported.</p> <p>A CAEModel item revision can have a relationship to the target product item revision. It can also have a relationship to a source CAEGeometry item revision or a source non-CAE item revision. A CAEModel item can participate in a hierarchical structure that defines a CAE structure. The datasets attached to the CAEModel item revision store the mesh representation of the assembly or the component.” (Siemens Product Lifecycle Management Software Inc, 2012)</p>
CAEAnalysis	<p>“Defines the type of analysis to be done and the solver used to perform the analysis.</p> <p>A CAEAnalysis item revision consists of solver parameters and the relationship to the CAEModel and CAEResult item revisions. The datasets attached to the CAEAnalysis item revision store the solver-specific input deck file and/or the tool-specific simulation file.” (Siemens Product Lifecycle Management Software Inc, 2012)</p>
CAEResult	<p>“Used for managing results of the CAE analysis from different solvers. The CAEResult item revision can have a relationship to the driving CAEResult item revision.” (Siemens Product Lifecycle Management Software Inc, 2011)</p>

Simulation items are used to manage simulation-focused files, data and information. The base for the simulation is often a geometrical description such as a CAD-model. These files, data and information are not managed by the simulation items. In Figure 4-3 the additional Item is called CAD item (Kondragunta, 2010). This graphic is also an example of the data management in TEAMCENTER. The CAD item or CAD item revision is provided by a designer. The monitored CAD model file is stored in the UGMASTER dataset under the CAD item revision. This CAD model file is the starting point of the simulation process. The CAD model is ideal to create an ideal CAD model for the later meshing. This CAD model is managed under the CAEGeometry item revision. In the case example, the designed CAD model under the CAD item revision has to be reduced. So, the CAD model of the CAEGeometry item revision

includes only a small section of the original data. This section is marked with a red ellipse in Figure 4-3. The reduced CAD model file is stored under the CAEGeometry item revision in a CAEGeom dataset. The CAD model of the CAE Geometry item revision is ideal for meshing. The mesh result is stored under the CAEModel item in the CAEMesh dataset. Additional load cases, boundary conditions and solver parameter definitions are necessary. These are stored under the CAEAnalysis item revision in the CAESolution dataset. With these, the generation of the bulk data deck and the solver input deck is possible. These files, data and information are also stored under the CAEAnalysis item in the CAESolution dataset or optional dataset types. With the solver input deck, the solve run is possible. The solve run produces multiple result and log files. These files are stored under CAEResult item in the CAESolution dataset. The resultant files of the solve run are visualised, interpreted and analysed. These files, data and information are also stored under the CAEResult item in the CAESolution dataset or optional dataset types.

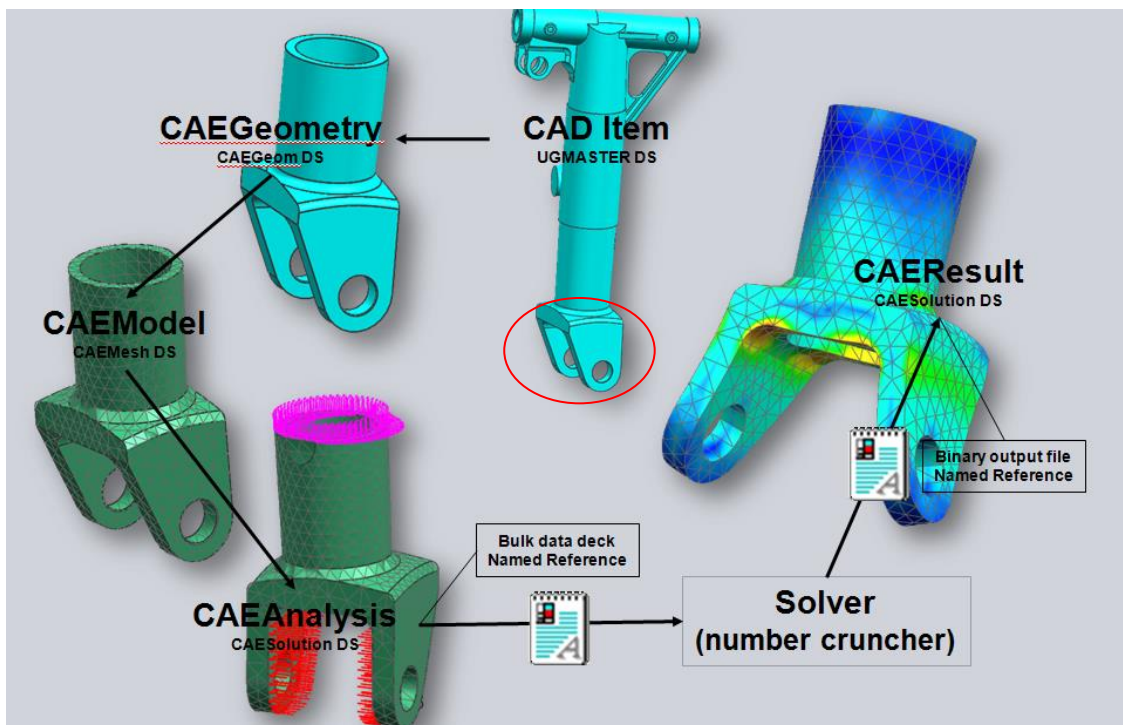


Figure 4-3 CAE Item Graphic (Kondragunta, 2010)

The simulation-data-management-structure is universal and also useable for manual analysis. An example is presented in Figure 4-4. Under the CAD item in the master dataset, a sketch of the product is stored in this example. Information on the sketch can be converted to the minimum required data such as dimensions, which are managed under the CAEGeometry item in the CAEGeom dataset. From the converted data, engineering data such as stiffness or section modulus can be calculated. The management of these files, data and information can be managed under the CAEModel item in the CAEMesh dataset. The documentation and



description of the analysis process, including answers to questions such as ‘How will the analysis be done and what equations, as well as the definition of the variables and parameters, will be used?’ These files, data and information can be stored under CAEAnalysis item in the CAESolution dataset. The calculated results or the files, data and information of the calculation (such as a MS Excel document) of the analysis and the interpretation can be stored under CAEResult Item in the CAESolution dataset.

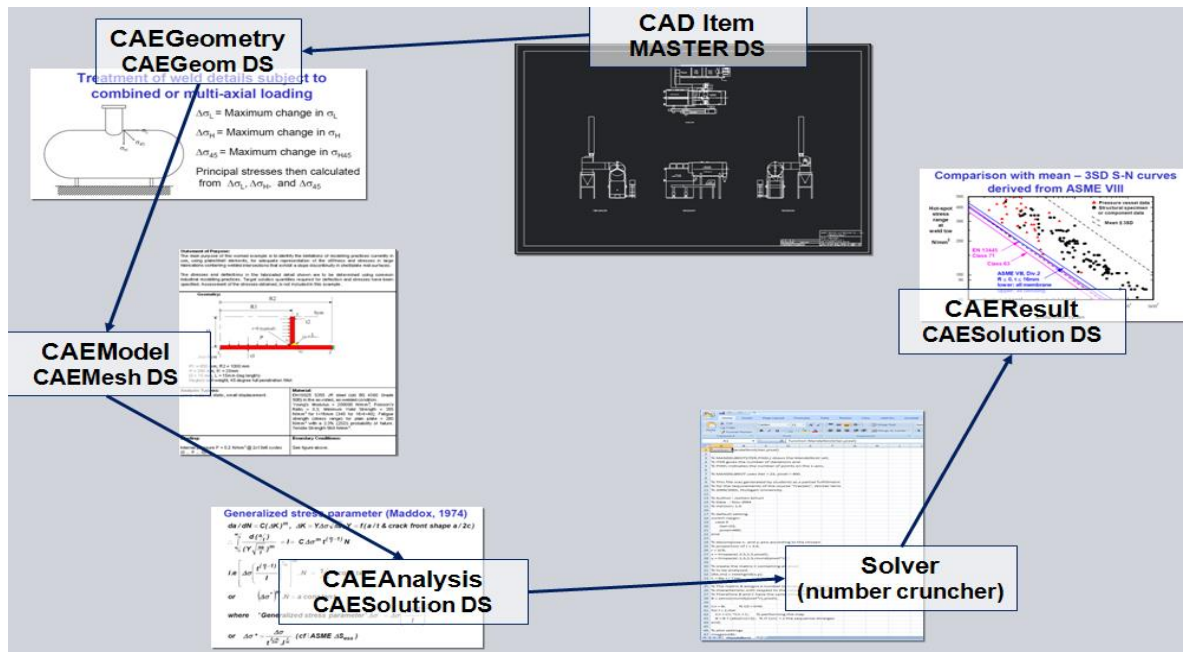


Figure 4-4 Manual Data Structure Example (Kondragunta, 2010)

The files, information and data change during the lifecycle. So, they have to be revisable. The revision technology of TEAMCENTER is presented in Section 4.3.

### 4.3 DATA ACQUISITION FROM THE REVISION OF CAE ITEMS

In the product lifecycle, the data changes through an on-going development process or a change process. In order to handle the change of the data, the items are sub-ordered into item revisions by the data management in TEAMCENTER (see Section 4.1). Changes can also influence the simulation. If a simulation is closed and the simulation input data changes to a new item revision, the analyst has to decide if the closed simulation is valid or if the simulation has to be revised. Therefore, all the CAE items are also organised in CAE items and CAE item revisions as in Section 4.1.

“CAE items are the fundamental workspace objects in TEAMCENTER used to manage CAE information. [...] CAE items are generally used to maintain the CAE representation of a product, part, or component. [...] CAE item revisions are workspace objects used to manage changes (revisions) to CAE items during the product design lifecycle.” (Siemens Product Lifecycle Management Software Inc, 2012)

The following is a list of CAE item classes and CAE item revision classes in TEAMCENTER 9.1:

CAE item – classes:

- CAEAnalysis
- CAEGeometry
- CAEModel
- CAEResult

CAE item revision classes:

- CAEAnalysisRevision
- CAEGeometryRevision
- CAEModelRevision
- CAEResultRevision

The ordering and structuring of the CAE items is similar to the normal ordering and structuring of the items. Figure 4-5 presents the ordering and structuring of the CAE items which is similar to Figure 4-2.

In most documents, the meaning of CAE item and CAE item revision is not separated from each other. For the most part, CAE item means CAE item revision. For example, the document of Kondragunta (2010) used as a source for the Figure 4-3 and Figure 4-4.

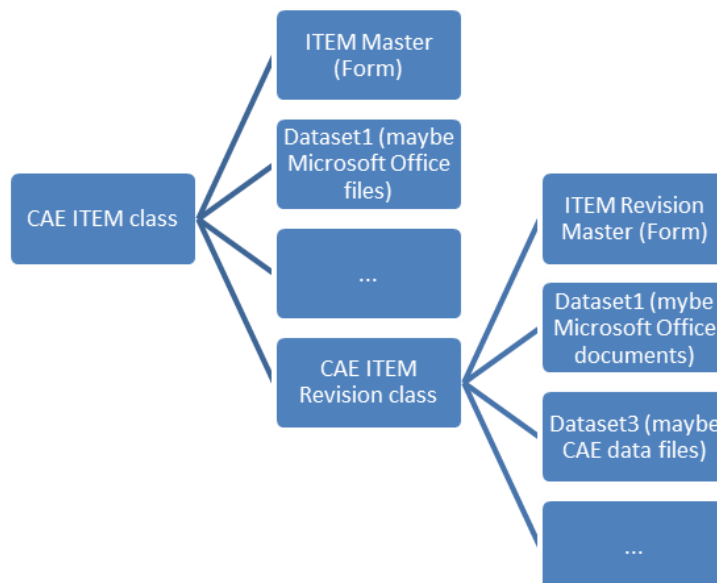


Figure 4-5 CAEItem Data Structure



The CAE item revisions do not directly include files. They are stored under datasets. So, the CAE item revision points to datasets, as discussed in Section 4.4.

#### 4.4 DATA ACQUISITION OF DATASETS TECHNOLOGY IN TEAMCENTER

The files, information and data belonging to a specific CAE item revision are stored in attached datasets. This is also discussed in Section 4.1 and monitored in Figure 4-5. In this section, some special dataset types will be presented. In the case of simulation data and process management with TEAMCENTER, five specific CAE datasets are available which are listed in Table 4-2 CAE Dataset Type (Siemens Product Lifecycle Management Software Inc, 2012).

**Table 4-2 CAE Dataset Type (Siemens Product Lifecycle Management Software Inc, 2012)**

<b>Dataset Type</b>	<b>Purpose</b>
CAEGeom	Contains a model geometry file.
CAESolution	Contains a simulation file.
CAESolver	Contains a solver-specific data deck.
CAEMesh	Contains a mesh file.
CAEResult	Contains the result files for a defined analysis.

Normally, the datasets are dedicated to the specific CAE item revision like the CAEGeom dataset to the CAEGeometry item revision or the CAEResult dataset to the CAEResult item revision. The datasets are not restricted to this kind of usability; other kinds of dedications are possible. For example in Figure 4-3 and Figure 4-4, the CAESolution dataset is dedicated to a CAEResult item revision.

The SDPM of TEAMCENTER is configurable and achieves a high level of flexibility in usability. It also includes a standard data model. In Section 4.5, this standard simulation data model will be described.

#### 4.5 DATA ACQUISITION OF THE STANDARD SIMULATION DATA MODEL OF TEAMCENTER

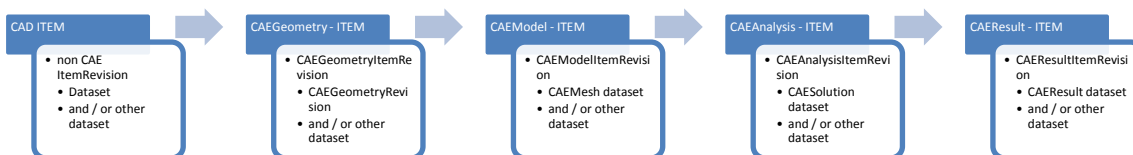
The high number of different simulation processes appearing on the market should be supported by the standard simulation data model of TEAMCENTER. In this section, a standard simulation process of a finite element analysis is listed. This example is similar to the simulation discipline neutral simulation process presented in Section 2.3.

1. A standard simulation process starts with CAD geometry. The CAD geometry data has to be managed via TEAMCENTER.
2. This CAD geometry was improved to make it ideal for meshing. The idealised CAD geometry data has to be managed via TEAMCENTER based on the CAD geometry data.
3. Based on the idealised CAD geometry, the mesh is generated and, as such the mesh data has to be managed via TEAMCENTER.
4. Load cases, boundary conditions and solver parameters are added to the mesh information. During this step, the analysis data is fully defined and solvable. The analysis data has to be managed via TEAMCENTER and based on the mesh data.
5. The solve run generates results and protocol files. The result and protocol data has to be managed via TEAMCENTER and based on the analysis data.

This simulation process is supported by the standard simulation model of TEAMCENTER. Item types manage the data. The numbering of the item types is directly related to the previously presented simulation process:

1. CAD ITEM
2. CAEGeometry ITEM
3. CAEModel ITEM
4. CAEAnalysis ITEM
5. CAEResult ITEM.

Each of these simulation items is sub-ordered into item revisions (see Section 4.3) and datasets (see Section 4.4). The standard simulation model is monitored in Figure 4-6.



**Figure 4-6 Standard Simulation Data Model**

The use of this standard simulation data model is flexible. For example, in some cases a geometrical idealization in front of the abstraction, such as meshing, is not necessary. In this case, the used simulation data model can be set in a way that the CAEGeometry ITEM level is ignored. In other cases, the simulation authoring tool may not support separate files during simulation process steps and simulation data model levels. Therefore, the standard simulation data model of TEAMCENTER can be customised to the required simulation data model. A third example is the case of NX Advanced Simulation, where the results and the analysis data are stored under the CAEAnalysis ITEM revision in the CAESolution dataset instead of storing the results under the CAEResult ITEM revision. In summary, the standard simulation data model of TEAMCENTER can support nearly all simulation disciplines and simulation tools. This is achieved by the configurability of the simulation data model in

TEAMCENTER. The configured simulation data model does not have to include all CAE ITEM levels or dataset levels provided by TEAMCENTER.

For the configurability, the relationships between the different CAE items play an important role, which will be discussed in Section 4.6.

## 4.6 DATA ACQUISITION OF THE RELATIONSHIPS TO ORGANISE THE SIMULATION DATA MODEL

In order to keep the simulation data models flexible, they should not be organised in a fixed or static way. The standard simulation data model in TEAMCENTER is represented in Figure 4-6. In order to achieve a dynamic and flexible simulation data model, the relationships between the CAE item levels are managed separately from the simulation data.

The relationships can be organised in three different use cases:

1. Relationships between the simulation data models
2. Relationships between CAE ITEM revisions and CAE datasets
3. Relationships between simulations

### 4.6.1 RELATIONSHIPS BETWEEN THE SIMULATION DATA MODEL

Relationships should represent the dependencies between item revisions. A relationship also includes the information relating to relation category. Two kinds of information are included in relationships:

- What item is dependent on the other item and
- Is it a data dependency or an organised dependency?

For the simulation data model in TEAMCENTER, seven relation types are available and are listed in Table 4-3 CAE Relation Type Source: (Siemens Product Lifecycle Management Software Inc, 2012).

Table 4-3 CAE Relation Type Source: (Siemens Product Lifecycle Management Software Inc, 2012)

Relation Type	Description	Primary Object	Secondary Object
CAE Criteria Relationship	Defines the relationship between a CAEModel ITEM revision and an ITEM revision that contains the criteria file.	CAEModel ITEM revision	Any item revision
CAE Defining Relationship	Defines the relationship between a CAEAnalysis ITEM revision and CAEModel ITEM revision.	CAEAnalysis ITEM revision	CAEModel ITEM revision
CAE Include Relationship	Defines the relationship between CAEAnalysis ITEM revisions to another. It indicates that a named reference in a CAESolver dataset associated to a CAEAnalysis ITEM revision contains and/or includes statements that refer to another named reference of a CAESolver dataset attached to a related CAEAnalysis ITEM revision.	CAEAnalysis ITEM revision	CAEAnalysis ITEM revision
CAE Parameter Relationship	Defines the relationship between a CAEModel ITEM revision and an item revision that contains the meshing parameter file.	CAEModel ITEM revision	Item revision containing the meshing parameter file
CAE Results Relationship	Defines the relationship between a CAEAnalysis ITEM revision and CAEResult ITEM revision.	CAEAnalysis ITEM revision	CAEResult ITEM revision
CAE Source Relationship	Defines the relationship between: <ul style="list-style-type: none"> <li>• A CAEModel ITEM revision and an ITEM revision representing product geometry.</li> <li>• A CAEGeometry ITEM revision and an ITEM revision representing product geometry.</li> </ul>	CAEModel ITEM revision or CAEGeometry ITEM revision	CAEGeometry ITEM revision or any item revision representing product geometry
CAE Target Relationship	Defines the relationship between: <ul style="list-style-type: none"> <li>• A CAEModel ITEM revision and an ITEM revision representing product geometry.</li> <li>• A CAEGeometry ITEM Revision and an ITEM revision representing product geometry.</li> <li>• A CAEAnalysis ITEM revision and an ITEM revision representing product geometry.</li> </ul>	CAEModel ITEM revision or CAEGeometry ITEM revision or CAEAnalysis ITEM revision	ITEM revision representing product geometry

The important relationships for the standard simulation data model are the CAE Target, Source, Defining and Result Relationship which are discussed in the following:

- CAE Target Relationship
  - This relationship should answer the question: “What Item (CAEGeometry ITEM revision or non-CAE ITEM revision) will these CAE items (CAEModel item revision or CAEGeometry item revision) represent?” This is from the perspective of the higher levelled simulation data model ITEM revision of the relationship.
  - This relationship should answer the question: “What CAE item (CAEModel item revision or CAEGeometry item revision) will represent this item (CAEGeometry item revision or non-CAE item revision)?” This is from the perspective of the lower levelled simulation data model ITEM revision of the relationship.
  - Additionally, this relationship should describe an organizational relationship, which describes the representation of a lower levelled item by a higher levelled item in the simulation data model. The relational points form a CAEModel item revision or CAEGeometry item revision to a non-CAE item revision or CAE Geometry item revision.
  - This relationship is used between the following items (the relational ordering will be from the first listed ITEM to the second):
    - CAEGeometry item revision and non-CAE item revision
    - CAEModel item revision and CAEGeometry item revision
    - CAEModel item revision and non-CAE item revision
- CAE Source Relationship
  - This relationship should answer the question: “What was the source item revision (CAEGeometry item revision or non-CAE item revision) to build up these CAE item (CAEGeometry item revision or CAEModel item revision)?” The perspective is from the higher levelled simulation data model ITEM revision of the relationship.
  - This relationship should answer the question: “What CAE item (CAEModel item revision or CAEGeometry item revision) is created based on this item (CAEGeometry item revision or non-CAE item revision)?” The perspective is from the lower levelled simulation data model ITEM revision of the relationship.
  - This relationship is a dependency relation and should describe the source of a CAEModel item revision or CAEGeometry item revision to a source input item.
  - This relationship is used between the following items (the ordering is from the first listed item to the second):
    - CAEGeometry item revision and non-CAE item revision
    - CAEModel item revision and CAEGeometry item revision
    - CAEModel item revision and non-CAE item revision

- CAE Defining Relationship
  - This relationship should answer the question: “What was the source of the CAEAnalysis item revision?” The perspective is from the higher levelled simulation data model CAEAnalysis item revision of the relationship.
  - This relationship should also answer the question: “What CAEAnalysis item revision will be created based on this CAEModel item revision?” The perspective is from the lower levelled simulation data model CAEModel item revision of the relationship.
  - This relationship is a dependency relationship and describes the source relation from a CAEAnalysis item revision to a source input CAEModel item revision.
  - This relationship is used between the following items (the ordering is from the first listed item to the second):
    - CAEAnalysis item revision and CAEModel item revision
- CAE Results Relationship
  - This relationship should answer the question: “What was the source of the CAEResult item revision?” The perspective is from the higher levelled simulation data model CAEResult item revision of the relationship.
  - This relationship should also answer the question: “What CAEResult item revision will be created based on this CAEAnalysis item revision?” The perspective is from the lower levelled simulation data model CAEAnalysis item revision of the relationship.
  - This relationship is a dependency relation and describes the source relation from a CAEAnalysis item revision to an result output CAEResult item revision
  - This relationship is used between the following items (the ordering is from the first listed item to the second):
    - CAEAnalysis item revision and CAEResult item revision

Most of the previous, theoretically described relationships are used in the following example:

The example represents an FEM simulation of a moulding tool. The geometrical description in the form of a CAD model is stored under the non-CAE item revision 000094/A. This non-CAE item revision is related with a CAESource and CAETarget relationship to the CAEGeometry item revision 000096/A. In the CAEGeometry item revision 000096/A, the reduced and idealised geometrical CAD model-file is stored in the CAEGeom dataset. This derivation is required to generate a geometrical CAD model that will be ideal for meshing. Therefore, this CAD model is derived from the non-CAE item revision 000094/A. The CAEGeometry item revision is related by a CAESource relationship to the CAEModel ITEM revision 000095/A. The mesh model is stored under the CAEModel item revision in the CAEMesh dataset. The FEM mesh model, in this example, does not include the load cases, boundary conditions and solver parameters. The CAEModel item revision is related to a CAETarget relationship, to the non-CAE item revision 000094/A. These CAETarget relationships monitor the representation

of the CAD model by the mesh model managed under the CAEMesh item revision. The CAEMesh item revision is related with a CAEDefining relationship to the CAEAnalysis item revision 000094.sim1/A. The load cases, boundary conditions and solver parameters and, in this example, the solver input deck, result and log files, are stored under the CAEAnalysis item revision in the CAESolution dataset. The relationship browser of TEAMCENTER helps to monitor the dependencies and make them easy to re-examine. An example is shown in Figure 4-7. The background is from the relationship browser of TEAMCENTER.

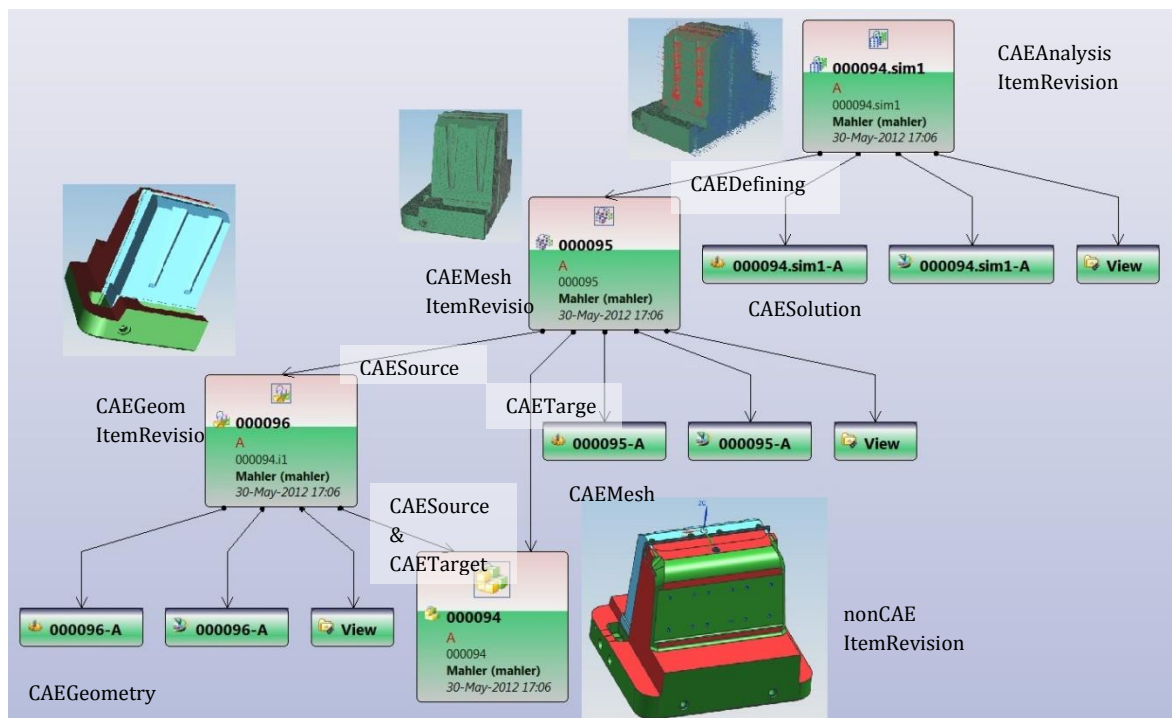


Figure 4-7 Simulation Data Structure Example

Relationship technology provides the possibility of a flexible, intelligent and efficient generation of standardised and individual simulation data models. Relationship technology is also used to relate items and item revisions to the datasets. This will be discussed in Section 4.6.2.

#### 4.6.2 RELATIONSHIP BETWEEN CAE ITEM REVISION AND CAE DATASETS

The relationship technology in TEAMCENTER is not only used to link item revisions to others but also to link the datasets to those items and item revisions. For example, in the case of linking a CAESolution dataset to a CAEAnalysis item revision, a CAESpecification relationship is used in the standard simulation model. The CAESpecification relationship points from the CAEAnalysis item revision to the CAESolution dataset. The CAESolution dataset can store multiple files. Each file managed in the dataset is linked by a reference type to the dataset.



With Reference type, a file can be dedicated to a specific functional type. Figure 4-8 shows an example where a Nastran bulk data file is managed, by the described technology, to manage datasets and relate them.

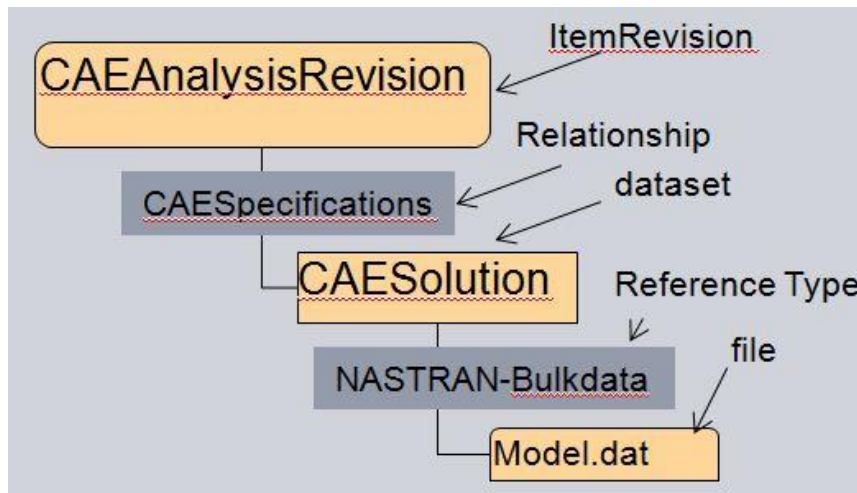


Figure 4-8 CAESpecification Relationship

The technology to manage datasets and relations of the datasets to items and item revisions does not influence the research project. However, to understand how TEAMCENTER manages and stores the files, this technology has to be discussed in order to understand the dataset technology without going into further detail. The relationship technology supporting the standard, individual simulation data model and the technology of the dataset relations must be established. The significant question is: How can TEAMCENTER support actual relationships between different simulations? This will be discussed in Section 4.6.3.

#### 4.6.3 RELATIONSHIP BETWEEN DIFFERENT SIMULATIONS

The relationship technology of TEAMCENTER makes the simulation data model flexible and configurable. This makes the management of different simulation models simple and supports multiple simulation authoring tools and means that multiple simulation models have to interact with each other because one simulation model will be dependent on another simulation model. So, the CAEInclude relationship type is available in TEAMCENTER (Siemens Product Lifecycle Management Software Inc, 2012). However, this CAEInclude relationship is not uncomplicated enough to support inter-disciplinary simulations with more than three dependent simulation models. The CAEInclude relationship only represents the dependency of a simulation model to another.

Therefore, the CAEInclude relationship should point from a CAEAnalysis item revision to another CAEAnalysis item revision. This relationship can be retraced from both CAEAnalysis item revisions. With the CAEInclude relationship, two CAEAnalysis item revisions can be



coupled. Multiple CAEInclude relationships are possible from a CAEAnalysis item revision. Figure 4-9 shows a CAEAnalysis item revision (000097/A) with two CAEInclude relationships to two other CAEAnalysis item revisions (000022.sim1/A and 000094.sim1/A).

Line	Description	Relation
000097/A;1-test		
+ 000022.sim1/A;1	000022.sim1	CAE Include Relationship
+ 000094.sim1/A;1	000094.sim1	CAE Include Relationship

Figure 4-9 CAE Include Relationship Example

The relationship technology and the simulation data model help to export both files and data. The export of the data is required in order to run a simulation or to work on the files. The relationships, items, item revisions and datasets help to identify the right files and data and export them. The technology for exporting files and data for on-going work from the TEAMCENTER database will be discussed in Section 4.7.

#### 4.7 EXPORT OF FILES AND DATA FOR EXTERNAL PROCESSES FROM THE TEAMCENTER DATABASE

In order to run or edit a simulation model, the files and data have to be exported from the database managed by TEAMCENTER. In this way, the files are provided to load them into the solver or simulation authoring tool. Therefore, the files are usually exported to a file directory. The simulation authoring tools and solvers can be forced by a routine to start and load the data from this file directory. So, working on the files and data is possible. Using simulation authoring tools and solvers, changes of the data and files or the generation of new files can be achieved. The files and data have then to be re-imported to the TEAMCENTER database. Therefore, TEAMCENTER checks the files and data and if necessary, the files and data are uploaded to the database. This import can be into an existing or a newly generated dataset of the item or item revisions. All this is achieved in a managed mode by TEAMCENTER.

Important for the export of the required files and data is their identification. The simulation data model (items, item revisions and dataset) as well as the relationship technology aid their identification. In order to organise the export, TEAMCENTER Simulation Process Management provides a framework called "Simulation Tool Configuration". The data export and import, as well as the external simulation authoring tool and solver execution, can be configured there. This framework will be discussed in the following section.

TEAMCENTER provides a framework called “Simulation Process Management” to configure external processes. This includes the configuration of data exports and imports as well as the execution of external processes such as pre-, post- or solve-processes. Siemens Product Lifecycle Management Software Inc. (2012) describes the framework as follows:

“Simulation Process Management provides a framework for the configuration and launching of external processes such as pre-, solver and post-processes. It allows you to:

- Define and organise specific tools to gather process inputs.
- Create TEAMCENTER objects to hold tool output.
- Import tool output into TEAMCENTER.
- Configure a simulation tool to specify an object (item, item revision, or dataset) in the data model as the object to hold the output data files.
- Define rules to navigate from the primary input object to the output object through a combination of relationships where the originating item revision is either the primary or secondary object of the relationship.
- Define a naming pattern for each of the objects (item, item revision, or dataset) that are created during the tool launch.”

In order to launch an external simulation process from TEAMCENTER, five steps are required:

- Identification of the data and files.
- Export of the files and data from the TEAMCENTER database to the external file directory.
- Force the launching of the simulation authoring tools or solvers, and often, also the load of the simulation data and files.
- Checking the file directory after data changes.
- If required, re-import of the data and files from the file directory to the TEAMCENTER database.

An example that should help to understand how TEAMCENTER supports such external processes:

In Figure 4-10 an example of CAE file export and import is given. This example is based on Kondragunta (2010). The example runs a batch solve with NX Nastran. TEAMCENTER forces this external process, provides the necessary data and re-imports the new generated files. The necessary file for the NX Nastran batch run is the input deck (\*.dat file). The file is required by the NX Nastran solver which gets its solve job and data from the input deck. This file is stored as a NASTRAN bulk data reference type in the CAESolution dataset of a CAEAnalysis item revision. TEAMCENTER is configured in a way that the launch of the NX-Nastran solve-process is only possible in the case of a selected CAEAnalysis item revision. In the case of other selections, a launch of the process is not possible. Due to this pre-selection of

a CAEAnalysis item revision, the identification of the required input deck file can be done in two steps. TEAMCENTER is configured to follow from a source object with a relationship (which can also be a reference) to a goal object:

1. Following the CAESpecification relationship from the pre-selected CAEAnalysis item revision to the CAESolution dataset. TEAMCENTER is configured to follow the **source object** CAEAnalysis item revision (which has to be pre-selected) down to a **relationship** which is a CAESpecification relationship that points to a **goal object** which has to be a CAESolution dataset.
2. Following the NASTRAN bulk data reference type from the CAESolution dataset down to the Nastran input deck file. TEAMCENTER is configured to follow the **source object** CAESolution dataset (which is identified in step 1) down to a **relationship** which is a NASTRAN bulk data reference type that points to a **goal object** which is the Nastran input deck file.

Using this pre-configuration of search regulations, TEAMCENTER provides a technology to identify the required files and data. TEAMCENTER can then export the identified file to a predefined temporary file directory.

TEAMCENTER can force NX Nastran to start by using a pre-created and pre-configured batch routine (Siemens Product Lifecycle Management Software Inc., 2011). The batch routine gets parameters from TEAMCENTER as inputs. In this example, two parameters are important. These parameters define the location of the exported file as well as the naming of the exported file. With the implementation of these parameters in the batch routine, the start of NX Nastran can be combined with the loading of the input deck (exported from TEAMCENTER).

After and during the solve-process done by the NX Nastran solver, additional files such as result- (\*.op2) and log-files (\*.log, \*.f06) (Siemens Product Lifecycle Management Software Inc., 2011) are generated. The \*.op2 and the \*.f06 file should be imported to the TEAMCENTER database. However, these files should be managed under a CAEResult item revision in a CAEResult dataset.

The reference types for the import files are:

- The Nastran\_output2\_binary – reference-type for the \*.op2 file and
- The Nastran\_result\_log – reference-type for the \*.f06 file.

TEAMCENTER can be pre-configured by defining a source object, relationship and goal object. The relationships flow from the source object to the last goal object identifies the position of the file that should be managed. Missing objects can be automatically generated by TEAMCENTER. In this example, four steps will be necessary:

1. The start will also be the pre-selected CAEAnalysis item revision. Following a CAEResult relationship from this CAEAnalysis item revision to the CAEResult item revision: TEAMCENTER is configured to follow the **source object** CAEAnalysis item revision (which is the pre-selection) down to a **relationship** which is a CAEResult relationship that points to a **goal object** which is a CAEResult item revision. TEAMCENTER is configured to generate new CAEResult items, CAEResult item revisions as well as the CAEResult relationships if these objects are not available.
2. Following the CAESpecification relationship from the CAEResult item revision down to a CAEResult dataset: TEAMCENTER is configured to follow the **source object** CAEResult item revision (identified in step 1) down to a **relationship** which is a CAESpecification relationship that points to a **goal object** which is a CAEResult dataset. TEAMCENTER is configured to generate a new CAEResult dataset and a CAESpecification relationship if these objects are not available.
3. Following the Nastran\_output2\_binary reference-type from the CAEResult dataset down to a NX Nastran result file (\*.op2): TEAMCENTER is configured to follow the **source object** CAEResult dataset (identified in step 2) down to a **relationship** (a Nastran\_output2\_binary reference-type) that points to a **goal object** (a NX Nastran result file (\*.op2)). TEAMCENTER is configured to generate a new Nastran\_output2\_binary reference-type if this object is not available, as well as to import the \*.op2 file located in the pre-configured temporary file directory.
4. Following the Nastran\_result\_log reference type from the CAEResult dataset down to a NX Nastran log file (\*.f06): TEAMCENTER is configured to follow the **source object** CAEResult dataset (identified in step 2) down to a **relationship** (a Nastran\_result\_log reference type) that points to a **goal object** (a NX Nastran log file (\*.f06)). TEAMCENTER is configured to generate a new Nastran\_result\_log reference type if this object is not available, as well as to import the \*.f06 file located in the pre-configured temporary file directory.

The configuration of TEAMCENTER achieves external process execution as well as data export of required files and data, and data import. The imported file \*.op2 is managed under the Nastran\_output2\_binary – reference type and the \*.f06 file under the Nastran\_result\_log – reference type ordered under the CAEResult dataset which is itself ordered under the CAEResult item Revision. This example can be seen in Figure 4-10.

In addition to the product “Simulation Process Management” integrated in TEAMCENTER, the technology of “Behaviour Models” is available. In the previous sections, the TEAMCENTER product “Simulation Process Management” was discussed. In Section 4.8, the TEAMCENTER “Behaviour Models” technology will be discussed.

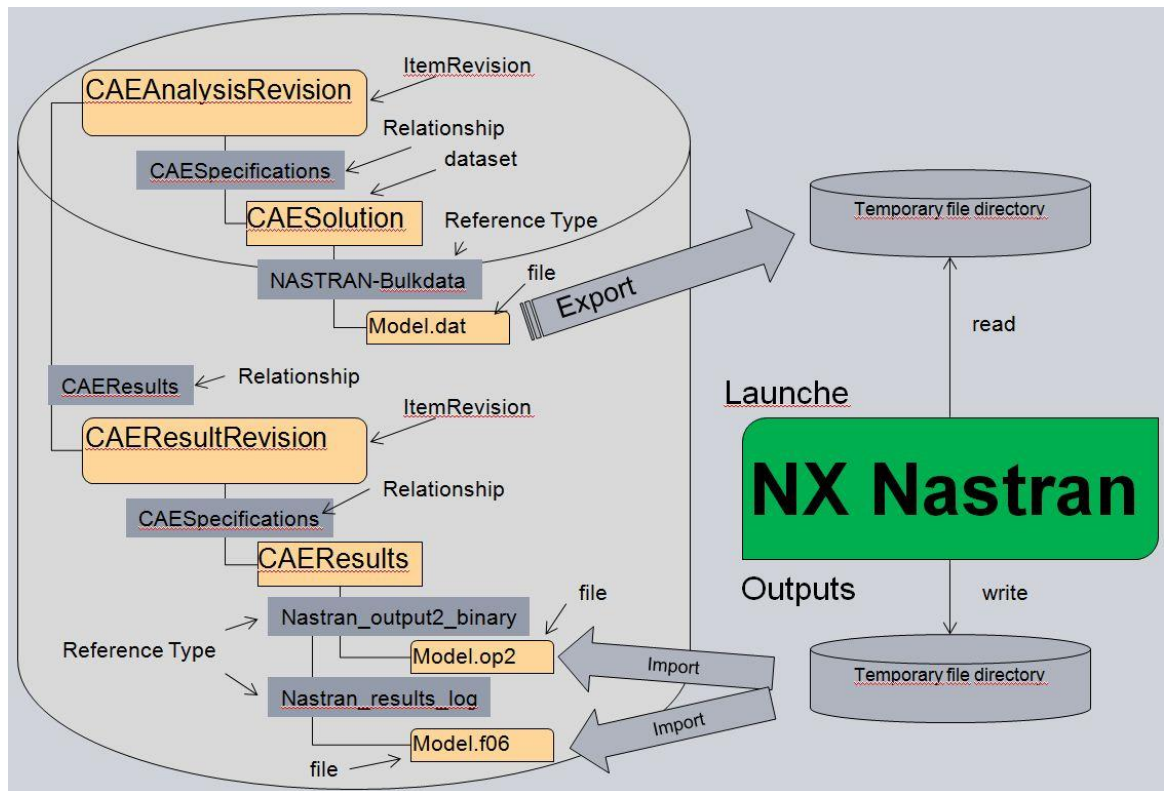


Figure 4-10 Data Export and Import

#### 4.8 DATA ACQUISITION OF THE BEHAVIOUR MODELS TECHNOLOGY

The Behaviour Models technology is not part of the TEAMCENTER “Simulation Process Management”. However, it is part of the “Mechatronics Process Management” product integrated within TEAMCENTER. With the TEAMCENTER “Mechatronics Process Management” product, the following point should be focused on:

“Siemens PLM Software believes that today’s complex [...] products require a systems-driven approach to product development that combines systems engineering with an integrated product definition [...]” (Siemens Product Lifecycle Management Software Inc., 2011)

In this sense, Siemens PLM Software (2011) declares that the TEAMCENTER “Mechatronics Process Management” product:

“establish[es] a collaborative environment for developing products comprised of mechanical, electronic, software and control (electrical interconnect) technologies[...] a common data model that crosses multiple engineering domains and a product lifecycle management (PLM) framework that manages the entire lifecycle process, these solutions

enable domain-specific teams to retain their mechanical, electrical, electronic or software focus while working together to meet overall product development goals.”

In other words, the Behaviour Models technology is focused to support a mechatronic development process. This will be discussed in Section 4.8.1.

#### *4.8.1 DISCUSSION: THE BEHAVIOUR MODELS TECHNOLOGY IN THE CONTEXT OF MECHATRONIC DEVELOPMENT PROCESSES*

Behaviour models represent the behaviour of functions. The behaviour model data management framework manages the behaviour models dependent upon the functions. Multiple behaviour models are dependent upon each other which is similar to multiple functions that are dependent upon each other. The summary of models and dependencies represent a system. The models represent the behaviour of sub-ordered functions. These dependencies are also managed by the behaviour model technology.

“The TEAMCENTER behaviour modelling tool integration framework is a generic integration framework and can integrate with any behaviour modelling tool [...] TEAMCENTER currently supports integration with the MATLAB Simulink tool.” (Siemens Product Lifecycle Management Software Inc., 2012).

Siemens Product Lifecycle Management Software Inc. (2011) discusses the requirement of technologies and products such as “Mechatronics Process Management” and “Behaviour Models”:

“The continued introduction of electrical, electronics and software components into the product development process has created the need for more efficient and effective integration of all participating engineering disciplines. Previously, much of this cross-domain knowledge was held in the heads of individual engineers. But that is no longer possible [...] because of that complexity, it takes a long time to be absolutely sure the [...] supply chain has covered everything and understands the impact that individual decisions are having on other aspects of the product design or its manufacturing processes.

This product complexity makes it difficult to reach the customer driven product requirements. New processes and structures have to be installed like:

“[a]dvanced modelling and simulation, the ability to derive engineering requirements from user needs and validate that your engineering specifications fulfil these needs early in the product development process [...] Complex software-driven electronics play a major role in many products’ most advanced features. To address the product development issues that arise from these complexities, TEAMCENTER’s suite of mechatronics process management solutions facilitates a collaborative environment that enables disparate engineering disciplines to work together as they develop products



comprised of multiple mechanical, electronic, software and electrical interconnecting components.” (Siemens Product Lifecycle Management Software Inc., 2012)

A theoretical example of a mechatronics system is pictured in Figure 4-11. This figure shows that the mechatronics system consists of mechanical and electronic, as well as software and controller disciplines. Also, sensors and the actuators are part in the mechatronic system. The volatility of the mechatronics system requires an information flow between the different elements and those different elements are developed by different departments.

“To address the product development issues that arise from these complexities, Mechatronics Process Management solutions facilitate a collaborative environment that enables different departments to work together as they develop products comprised of multiple mechanical, electronic, electrical, and software interconnecting components.” (Siemens Product Lifecycle Management Software Inc., 2012).

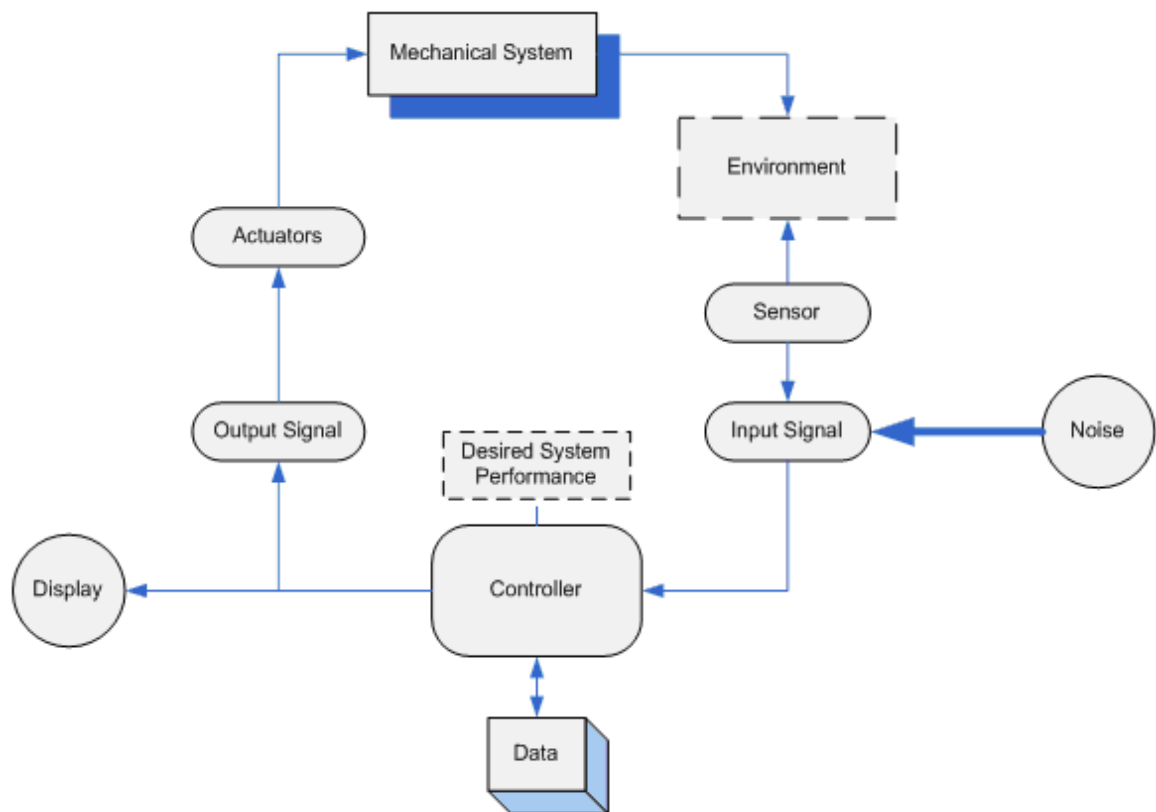


Figure 4-11 Mechatronics System (Siemens Product Lifecycle Management Software Inc., 2012)

Therefore, TEAMCENTER supports mechanical design integration, electrical design integration, software design integration as well as wire harness design integration.

“To enable you to manage and control your source code development assets, TEAMCENTER integrates with IBM Rational ClearCase. Equally important, TEAMCENTER

provides best-in-class support for signal/message management, calibration and configuration parameter management, as well as software design component management.” (Siemens Product Lifecycle Management Software Inc., 2012)

The behaviour model theory is also part of the support of a mechatronic development process. TEAMCENTER, together with the MATLAB/Simulink simulation authoring tool, can support the behaviour model theory. Poldermann & Willems (1998) mentioned the following about behaviour equations:

“We therefore speak of behavioural equations when mathematical equations are intended to model a phenomenon. It is important to emphasise already at this point that behavioural equations provide an effective, but at the same time highly non-unique, way of specifying behaviour. Different equations can define the same mathematical model.”

It follows that a behavioural equation recognises a result as output data in relation to input data. This mathematical equation can be integrated into a mathematical model by using an equation based simulation (EBS) tool. These mathematical models are called behaviour models because they represent the behaviour, for example, of functions, products or systems.

Giese, Graf & Witz (1999) discuss the benefits of the behaviour models:

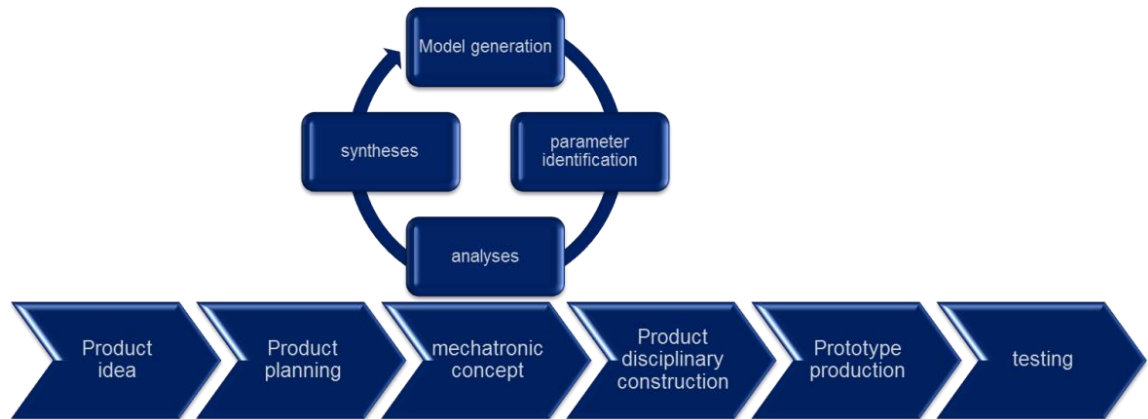
“Today’s software systems for business, telecommunication and industry often obtain a high inherent complexity concerning their structure and behaviour. Their development demands construction techniques like multi-layer architectures, fine grain class structures, distribution, concurrency, reactivity, etc. to meet their requirements and change over time. The resulting software architecture has to support maintenance and configuration aspects. The object-oriented modelling principle allows to abstract and (de)compose system properties systematically. It offers tools to transform these properties into appropriate object-oriented structures and behaviour. [...] Its rich set of notations allows to express system requirements, system structures and behaviour independent from any specific software development processes.”

This shows that the behaviour model does not need the exact solution of the phenomena. In order to build a behaviour model of a software solution, the software code of the solution is not required; it is enough to know the phenomena that result in the solution. The behaviour model can be generated much earlier in the development process than the software code. The behaviour models can be used to detail the software code of the phenomena during the development process.

The behaviour models can also support the model-based mechatronic development methodology. Based on a design-loop, an optimization of the mechatronic concept can be realised at an early stage of the development process (Lennon, 2007; Klotzbach, Oedekoven & Grassmann, 2011). The model-based development process integrates a holistic approach in this early phase. In front of the discipline-oriented development phases, a holistic mechatronic system phase is implemented. This holistic mechatronic concept phase works out a pre-optimised system-concept based on the behaviour models. The behaviour models



are at a high abstract level representing the function but not the constructive solution. The constructive solution is worked out later in the development process. Based on this mechatronic concept phase, a verification of the system concept is achievable. If the check seems to be positive, the next step of the development process can be started.



**Figure 4-12 Model-Based Mechatronic Development Methodology (Mahler, Framework for System-Simulation based System Engineering Development Methodology, 2012)**

The model-based mechatronic development methodology has progressed to a model based system engineering development methodology. The behaviour models used in the context of the model-based mechatronic development methodology are also useable in the context of the model-based system engineering development methodology. This will be discussed in Section 4.8.2.

#### *4.8.2 DISCUSSION: THE BEHAVIOUR MODELS TECHNOLOGY IN CONTEXT TO SYSTEM ENGINEERING*

Combining behaviour models with the model-based mechatronic development methodology is an effective step to reduce errors in hardware and software at late development stages, which create costly delays, and save development time (Lennon, 2007; Klotzbach, Oedekoven, & Grassmann, 2011). As such, an improvement to the model-based mechatronic development methodology can be achieved.

Stark, Beier, Wähler & Figge (2010) discuss the system engineering methodology:

“The system engineering process is a comprehensive, iterative and recursive problem solving process which is suitable for the development of mechatronic products. [...] First step is the analysis of customer requirements in order to derive functional and performance requirements. Subsequently, functions are identified, decomposed and allocated to the requirements. During synthesis the product is defined in terms of physical and software elements and afterwards verified against the requirements [...]”

Mahler(2012) discusses the improvement of combining the model-based mechatronic development methodology with the system engineering methodology:

“The model-based mechatronic development methodology and system engineering will get married together to model-based system engineering (Eigner, Gilz, Hollerith, & Zafirov, 07. Nov. 2011). The model idea of model-based mechatronic development methodology (like CAD models or executable models like equation based models) will help to verify the requirements generated in the system engineering development methodology. This check of the requirements against an early development stage will identify miss functions.”

This means that behaviour models will be used for verification of the development process by checking the product concepts or constructive solution during the development process.

“Behavioural models [...] play a critical role in cascading customer needs down into engineering requirements. For example, a customer-driven need might be to reduce vibration during engine idling. [...] How then can the customer's need for a smooth idling experience be translated into engineering requirements for the suspension and the suspension bushings? Systems-level models can play a very useful role in this process. Models can be built to capture the frequencies of the main subsystems.” (Siemens Product Lifecycle Management Software Inc., 2011)

The case examples of behaviour models can begin with an early development level or state. For example, functionalities of systems or subsystems can be modelled within the behaviour models (Oliver, Kelliher, & Keegan, 1997). Therefore, these behaviour models can exclude the technical solution to achieve this functionality. Multiple behaviour models representing functions can be linked to each other to represent the community of the system by simulating their functions. In this way, the function of the system concept can be checked without deep knowledge about how to solve the functionality. For example on the system level, a solution to achieve the function of the system is worked out. Behaviour models can be used to compare the mathematical abstraction of different possible function solutions. Also, verification and validation are possible with behaviour models because their simulation results are comparable with the defined needs and requirements of the simulated system, subsystem or component. The integration of behaviour models into the system engineering methodology creates the model based system engineering methodology. Brown & IBM Corporation (2011) mention that the integration of behaviour models into system engineering provides improved possibilities to validate and verify the development process or product. The significant issue is in understanding the validation and verification of the process due to the different and, consequently, complex perception of document phraseology. Nevertheless, behaviour models help to improve the development process as well as to support verification and validation.

The case example of the behaviour models in the system level of system engineering is similar to the case example of the mechatronic concept level in the model based mechatronic development methodology described in Section 4.8.1. Notably, in this early development stage, simulations cannot be CAD-based because the CAD models are not yet created or

detailed enough. The representation of the product has to be non-CAD-model-based but represent the behaviours. The behaviour representation of functions or abstracted constructive solutions is based on mathematical descriptions. The deeper integration of mathematical abstraction and formulation requires a mathematical equation based simulation (EBS). Today, the use of EBS models begins with the generation of system representing behaviour models at system-level (Lennon, November 21, 2007). "As a simple illustration, consider the use of spread sheet software" (Brown & IBM Corporation, 2011). Spread sheets can be used in a smart way because big systems require enormous manual work to model all the formulae.

TEAMCENTER supports the behaviour model technology, independent of case examples or the development process methodology. The behaviour model technology will be discussed in Section 4.8.3.

#### *4.8.3 DATA ACQUISITION OF THE BEHAVIOUR MODELS TECHNOLOGY IN TEAMCENTER*

In order to support the behaviour model theory, TEAMCENTER provides a technology for the data management of the behaviour models. The modelling of the behaviour models is achieved using EBS authoring tools. Currently, an interface for the simulation authoring tool Matlab/Simulink is standardised.

"With the integration of behaviour modelling tools with TEAMCENTER, you can use the behaviour modelling tool for model authoring and TEAMCENTER for model management. [...] The TEAMCENTER behaviour modelling tool integration framework is a generic integration framework and can integrate with any behaviour modelling tool. [...] TEAMCENTER currently supports integration with the MATLAB Simulink tool" (Siemens Product Lifecycle Management Software Inc., 2012).

The behaviour model technology of TEAMCENTER provides special objects of item and item revision as well as an interface, connection, relation and occurrence note type (Siemens Product Lifecycle Management Software Inc., 2012). The behaviour ITEM class is called a behaviour model item and includes relationships of a model file from an EBS authoring tool.

During the design lifecycle, files and data change. In order to manage changes to the files and data, revision technology is used (see Section 4.1). If the simulation model attached to the behaviour model item is edited and the simulation model file is changed, the analysts must decide if the edited behaviour model file has to replace the file of the behaviour model item or behaviour model item revision, or if the behaviour model item revision itself has to be revised. For the revision, all the behaviour model items are organised in the behaviour model item revision.

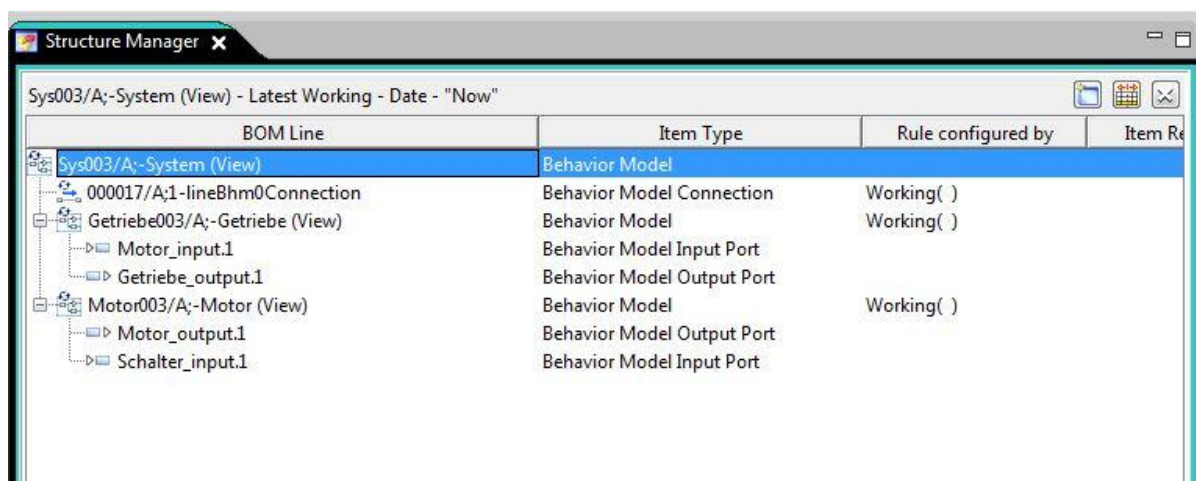
The behaviour model attached to the behaviour model ITEM revision can require input as well as output data. The interface requirement of the input and output data is represented as ports related to the behaviour model item revision. Two port types are available (Siemens Product Lifecycle Management Software Inc., 2012):

- Behaviour Model Input Port (Object Type: Bhm0InPort) representing the input interface of the model
- Behaviour Model Output Port (Object Type: Bhm0OutPort) representing the output interface of the model

The ports provide data about ‘What input as well as output ports are required?’ In order to view the input and output information flow between different item revisions, the ports have to be connected to each other. Information about the connection of the ports between different behaviour model item revisions is managed by the behaviour model connection (Object type: Bhm0Connection). The behaviour model connection points from a behaviour model output port to a behaviour model input port.

The linking of multiple behaviour models with behaviour model connections requires a hierarchical structure of the behaviour model item revision. In the case of CAD assemblies, in addition to the dataset, a data-structure including the subordinated behaviour model item revisions are stored (Siemens Product Lifecycle Management Software Inc., 2012). This is called bill of material view (BOM view).

The ports of the behaviour model ITEM revision are stored under the BOM view as behaviour model input port or behaviour model output port. Also, the behaviour model connection, linking two behaviour model ports, is stored under the BOM view. However, the behaviour model connection only connects ports of subordinated behaviour model item revisions, i.e. the connecting of ports is not possible on the behaviour model item revision level where the port is created. The earliest opportunity to connect ports is one level higher in the BOM view than the behaviour model item revision that provides the port for the connection. An example is shown in Figure 4-13.



BOM Line	Item Type	Rule configured by	Item Re
Sys003/A;-System (View)	Behavior Model		
000017/A;1-lineBhm0Connection	Behavior Model Connection	Working( )	
Getriebe003/A;-Getriebe (View)	Behavior Model	Working( )	
Motor_input.1	Behavior Model Input Port		
Getriebe_output.1	Behavior Model Output Port		
Motor003/A;-Motor (View)	Behavior Model	Working( )	
Motor_output.1	Behavior Model Output Port		
Schalter_input.1	Behavior Model Input Port		

Figure 4-13 Ordering of the Behaviour Data Model

With the integration of input and output ports as well as with the port connections between behaviour model item revisions, it is possible to create horizontal and vertical structured model-based system engineering models. The horizontal structure connects models at the same level and the vertical structure connects the subordinated models. The data model of the behaviour models overfills the recursive modelling of model-based system engineering as described in Brown & IBM Corporation (2011). An example is shown in Figure 4-14.

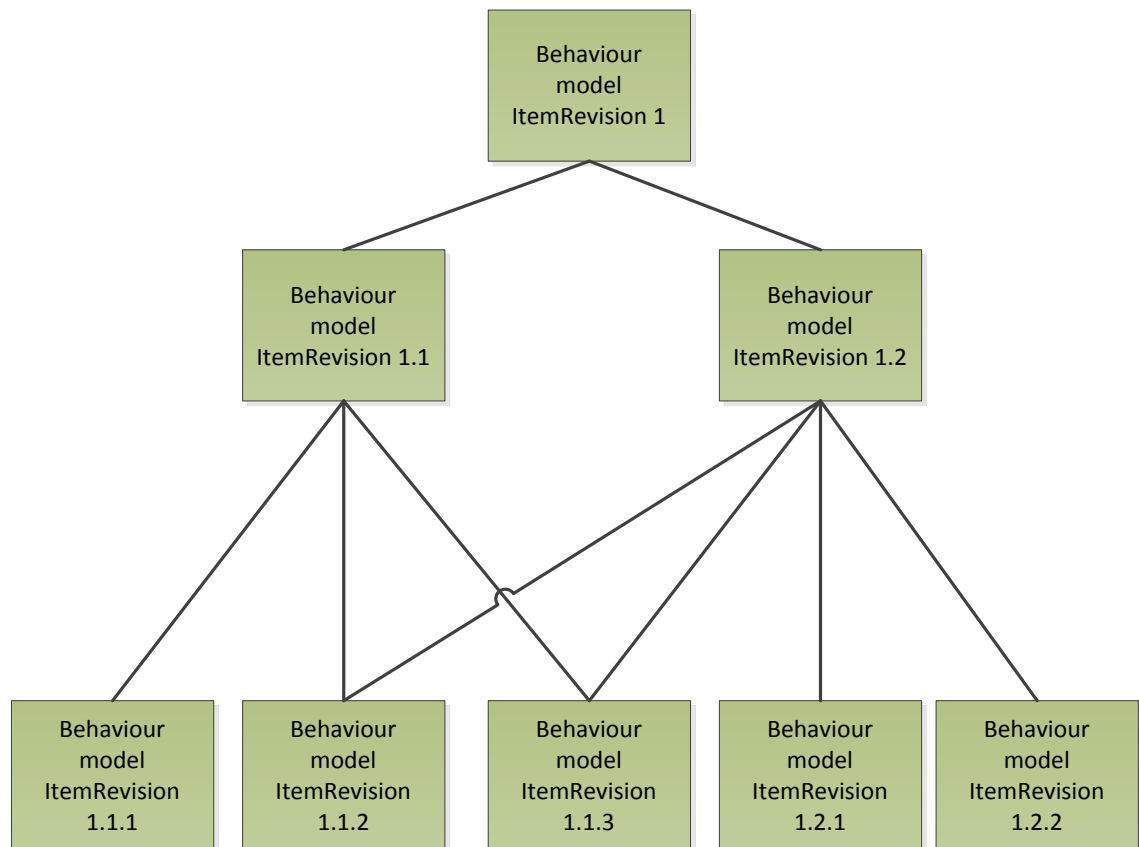
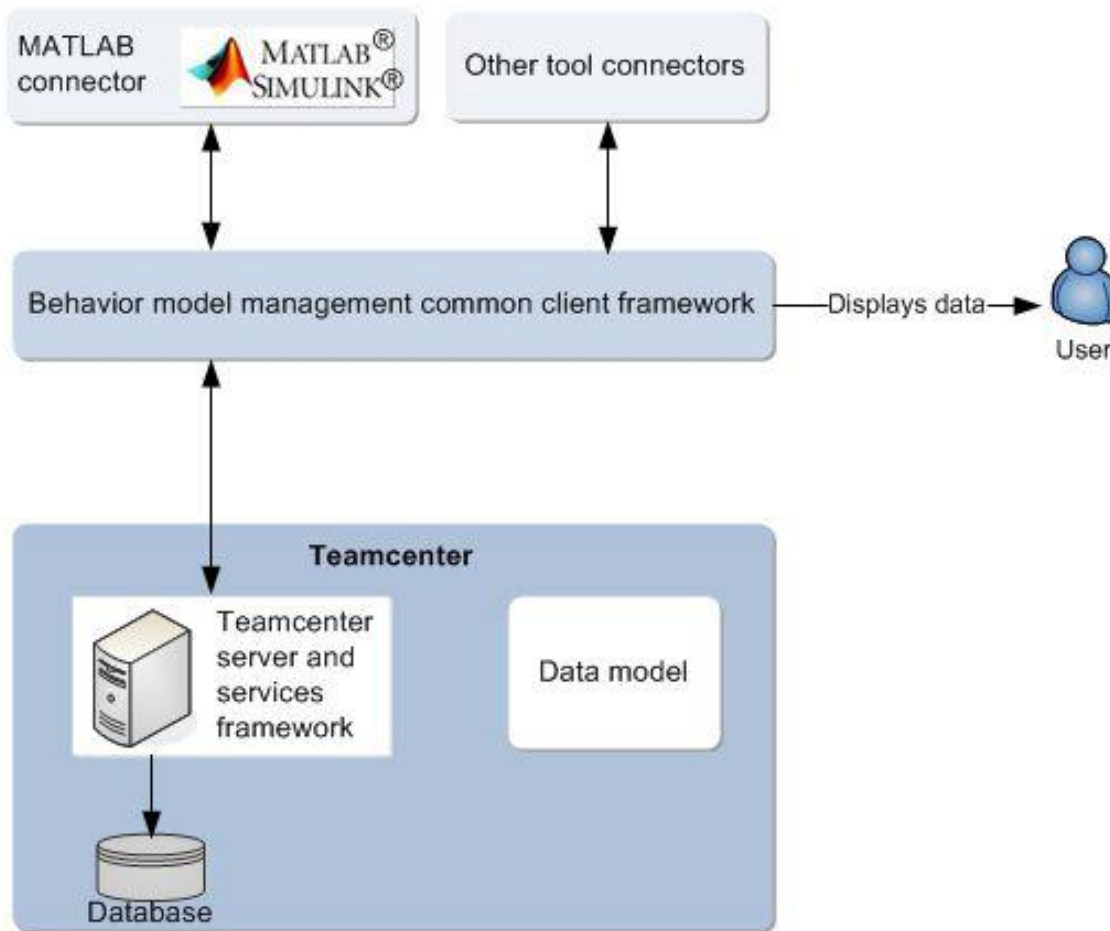


Figure 4-14 Structure of Model-Based System Engineering Models

The generation of the behaviour data models requires a mutual cooperation between the behaviour model authoring tool and TEAMCENTER. This integration is achieved between Matlab/Simulink and TEAMCENTER. The interface of Matlab/Simulink is set up through a behaviour model management common client framework. Therefore, TEAMCENTER is connected to the behaviour model management common client framework and the behaviour model management common client framework is connected to the Matlab/Simulink connector (see Figure 4-15). With this interface architecture, a generic integration framework is reached. Currently, only the interaction between TEAMCENTER and the simulation authoring tool Matlab/Simulink has been achieved.



**Figure 4-15 Matlab/Simulink Behaviour Model Interface Architecture (Siemens Product Lifecycle Management Software Inc., 2012)**

In order to manage the files, dataset technology is used. Datasets are attached to behaviour model items or behaviour model item revisions. In the case of global information, the dataset is attached to the behaviour model item and, in the case of changing information, the dataset is attached to behaviour model item revisions.

“The Dataset object represents an actual data file on the operating system or in TEAMCENTER. Datasets are typically authored content of some sort, such as Microsoft Office files or CAD data files.” (Siemens Product Lifecycle Management Software Inc., 2012).

The Matlab/Simulink simulation behaviour model file is stored as a dataset under the behaviour model item revision. The datasets are linked to the behaviour model item revision by using Specification relationships. This technology is similar to the CAESpecification relationship discussed in Section 4.6.2. The behaviour data model of TEAMCENTER is shown in Figure 4-16.

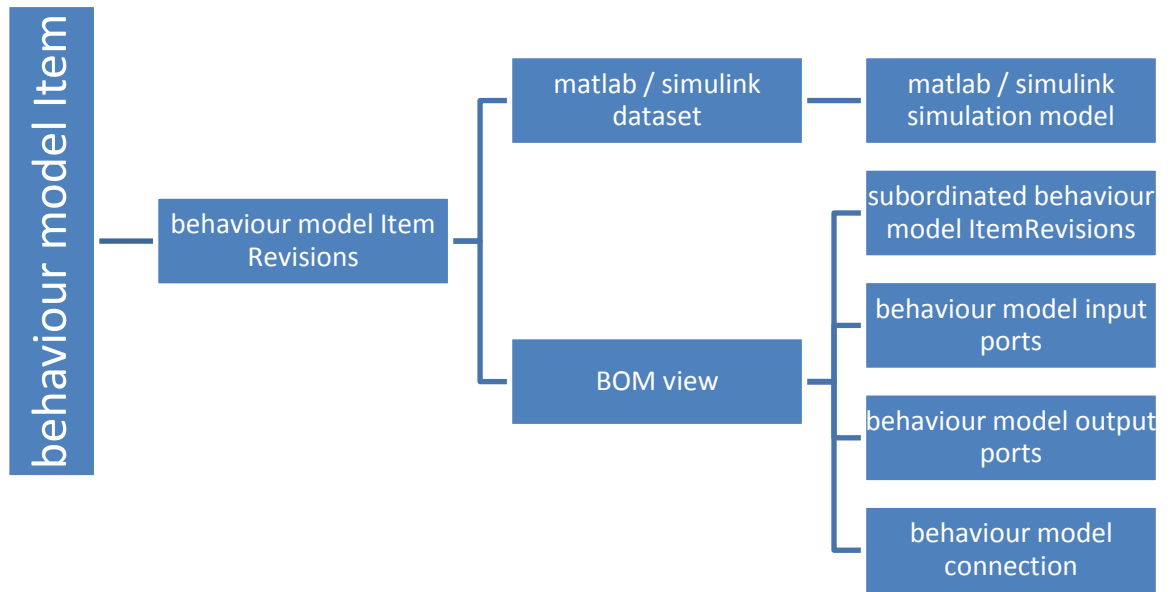


Figure 4-16 Behaviour Model Item Structure

The data acquisition of the simulation data management technology and behaviour model technology should help to understand and analyse case studies. These case studies will be discussed in the Chapter 5.



## 5 CASE STUDIES OF MANAGEMENT OF THE DATA OF MULTI-DISCIPLINARY AND INTER-DISCIPLINARY SIMULATION

As discussed in Chapter 2 and Chapter 5, the current state of simulation data management should be improved to achieve a multi-disciplinary simulation data management. The case study methodology should help to compare multi-disciplinary and inter-disciplinary simulation examples with the previous data acquisition of the simulation supporting technologies in TEAMCENTER. So, multi-disciplinary and inter-disciplinary simulation examples will be sampled. The examples are multi-disciplinary simulations whereby the data is managed with TEAMCENTER. These examples are analysed from the viewpoint of multi-disciplinary simulation data management. The data sampling method of the multi-disciplinary simulation examples is described in Section 3.2.2.

Each case study should be worked out and discussed based on the same process. This process should make the work easier and reviewable. The process is described as follows:

1. The first step is the data acquisition and description of the multi-disciplinary simulation application example. This will make the example comprehensible.
2. In the second step, the simulation process is detailed which should include the detailing of the simulation process and the simulation models and steps.
3. Then, in the third step, the files, data and information, used or generated during the simulation process, are detailed and mapped over the simulation process. Therefore, the case study examples are analysed from the viewpoint of information and data flows. This is done for each case and multi-disciplinary simulation example.
4. In the fourth step, the previously acquired and analysed files, data, information and data/information flow is mapped in/with the TEAMCENTER data management system. This will map the case study onto the actual state of simulation data management technology. Missing functionalities, data management objects or workarounds should become identifiable.

The case studies will use current available software. Such software will be used to generate, edit and manage the data. Authoring software such as NX could be used to generate and edit the data. This kind of software will include pre-processing, post-processing and solver functionalities. The data management software will be TEAMCENTER. TEAMCENTER is a leading technology that includes SDPM (see Section 2.6 till 2.9). TEAMCENTER seems to be the best system for the data collection for the proposition of the new framework. Nevertheless, all sorts of software will be used to generate, edit or manage the data. The generation, editing and managing of this data will be necessary in order to process implementation into the multi- and inter-disciplinary simulation case study examples. These case studies will provide the basis for the new scientific knowledge. Such new knowledge will be generated in the architecture, as well as ordering and relating rule sets for the new framework to manage and support multi- and inter-disciplinary simulation. Therefore the



new knowledge will help to generate an improved way for ordering and relating data, generated by current available authoring software. The current available data management software by TEAMCENTER will provide the leading software for data management and a basis for the case studies. TEAMCENTER also provides an implementation environment for improved data ordering and relating. Nevertheless, the goal will be to improve the work on multi- and inter-disciplinary simulation by generating new scientific knowledge for an improved architecture, encompassing ordering and relating rules for the multi- and inter-disciplinary simulation support and management. Therefore the case studies analyse the work and its processes on and off multi- and inter-disciplinary simulations.

In Section 3.2.2, four case studies and multi-disciplinary simulation examples have been discussed. The four examples that will be discussed in the next four sections are:

- 1) Example 1: SCHMIDT Gesellschaft für Werkzeug- und Formentechnik mbH; Nürnberg:  
Creation of a moulding tools simulation vision for moulding tool optimization.
- 2) Example 2: Semi-Finished Goods Producer, (unpublished customer name), South-Germany  
Project: Benchmark of simulation tools. This benchmark should identify an ideal simulation tool. An improved development process of extrusion sections production tools is achievable with Siemens simulation products.
- 3) Example 3: Automotive Company (unpublished customer name), South-Germany  
Project: The one-vendor-benchmark project. In this project, a multi-disciplinary and mechatronic system-simulation of a car luggage door system had to be realised.
- 4) Example 4: Automotive Company (unpublished customer name), South-Germany  
Project: A research project called "Interdisciplinary Model-based Development Process". This research project includes a multi-disciplinary and mechatronic system-simulation of a car windows lifter system.

For most of the examples the customer name will not be published. Indeed the publishing of the customer name could be in breach of the compliance regulations of the researcher's employer.

In Sections 5.2 through 5.5, four case studies including multi- and inter-disciplinary examples, will be discussed. A preliminary introduction into standard simulation process workflow will be discussed in Section 5.1.

## 5.1 SIMULATION PROCESS BASICS

A simulation process usually has three steps. The first step is called pre-process and creates a simulation model. This process deals with the opening or importing of input files and brings them into a simulation model that is runnable by a solver. The second step is called solve-process. In this step, the simulation model obtained from the first step is used by a solver that

executes the mathematical procedures of the simulation and produces the results. Meanwhile, solver-run-files are generated and are called log-files that record the process of the solver. The third step is called post-process. Here, the results generated by the solver, are graphically presented. These visualisation techniques are used to aid the analysis and interpretation of the results. Usually, the simulation is documented (Roensch, 2010). This process will be shown in the Figure 5-1.

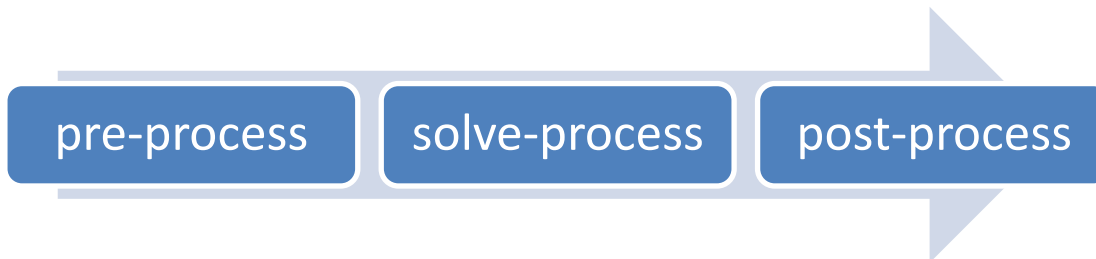


Figure 5-1 Basic Simulation Process

In Sections 5.1.1 through 5.1.3, these three processes will be discussed in detail.

### 5.1.1 PRE-PROCESS

The pre-process abstracts the objects or processes that should be simulated to a solver understandable format. Each solver has its specific mathematical base which defines the kind and type of abstraction (Dehning & Wolf, 2006, p. 8). So, the pre-process stage can include multiple process steps. The steps and number of steps are dependent on the kind and quality of the pre-process input data. The more modern approach is to use geometrical CAD data as input (Roensch, 2010). Input data can also be mathematical dependencies or simply an idea. Mathematical dependencies often appear in the case of equation based simulation (EBS). An idea can be modelled directly into pre-process because the documentation of the idea is not in a useable format here. Usually, specialists in geometrical CAD data inputs have to prepare this data to be ideal for the abstraction process (Roensch, 2010). Afterwards, the data can be transformed to the abstracted format. For a better understanding of abstracted formats, the following cases are given as examples:

- For an FEM simulation, the abstraction will be a mesh (with nodes and elements). The mesh represents finite elements (Roensch,2010). Then, the finite elements are transformed by the solver into a matrix. The matrix is used in the solve-process to generate results.
- For a CFD simulation, the abstraction will be a mesh. The mesh represents volume elements (Roensch,2010). Then, the volume elements are transformed by the solver into a matrix. The matrix is used in the solve-process to generate results.

- For an MBS, the abstraction will be object-linked information (such as drill point and axe) and parameters (such as velocity), representing links between bodies, bodies representing information (such as mass or volume) and parameters (Kecskemèthy, 31/2007), representing the geometrical bodies. Afterwards, the link and body information and parameters are transformed by the solver into a matrix. The matrix is used in the solve-process to generate results.
- For an EBS, the abstraction will often be an object-oriented, mathematical formulation of the dependencies or workflows (Zimmer, 2010) and the parameters for the equations. Afterwards, the mathematical formulations are used by the solver to generate results.

The abstraction of the pre-process input data requires a high number of parameters and variables describing the simulated process or product. Roensch (2010) discusses the necessity of parameters or variables, for example, parameters could be used to describe the elasticity as materials parameters, friction parameters, damping parameters or spring parameters. Additionally, the definition of load cases and boundary conditions has to be generated during pre-process.

A summary of a basic pre-process workflow is represented in Figure 5-2 Basic Pre-Process

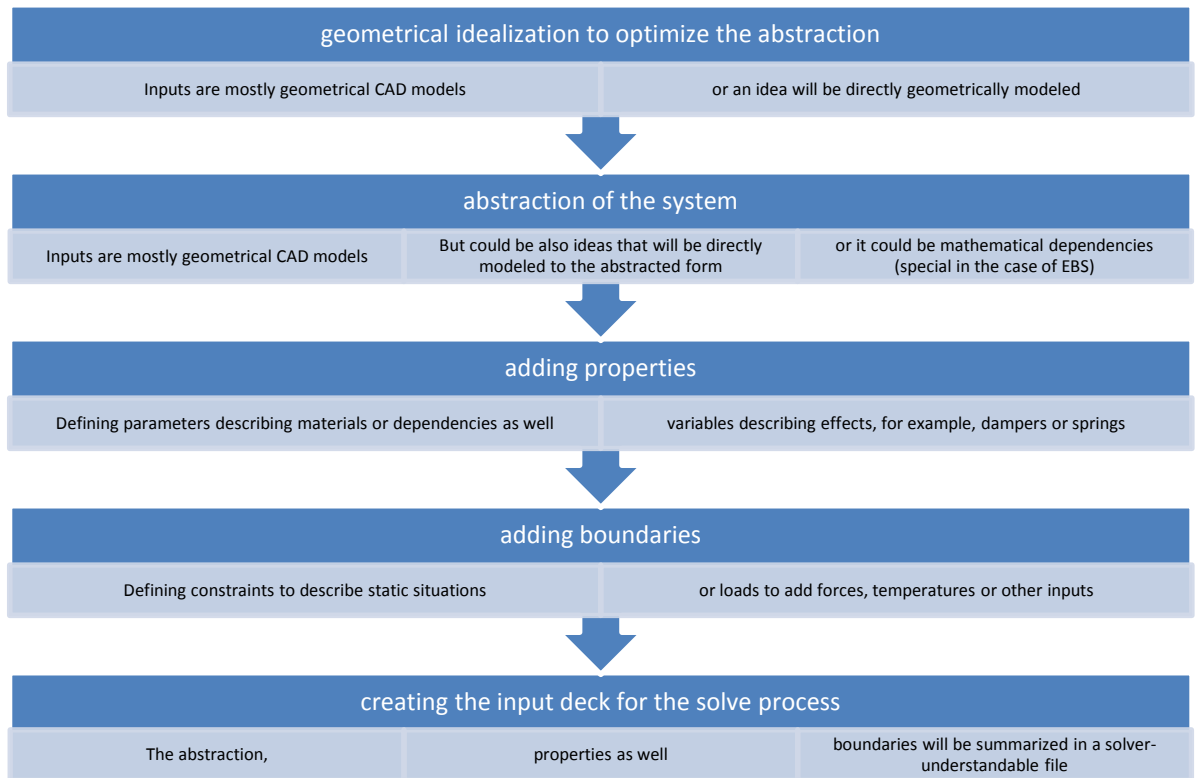


Figure 5-2 Basic Pre-Process Workflow

This basic pre-process is dynamic and can be changed to attain an optimised solution. The generated files are dependent on the optimised pre-process-workflow and the used simulation authoring tool (pre-process simulation tool).

In most simulation authoring tools (pre-process simulation tools), the solver is not directly integrated. However, the solve-process can be executed directly by the pre-process simulation authoring tool. In these cases, the simulation authoring tool has to generate an input deck for the solver. This input deck is a summary of the model describing abstraction and the parameters, variables, constraints and loads. The input deck has to be in a solver understandable format. The data of the input deck is required for the mathematical computation in the solver. For example: In the case of the SIEMENS product NX Advanced FEM, the pre-process workflow is handled as described above (Siemens Product Lifecycle Management Software Inc., 2011). This structure is the same in the case of an FEM or CFD simulation. The example is shown in the Figure 5-3.

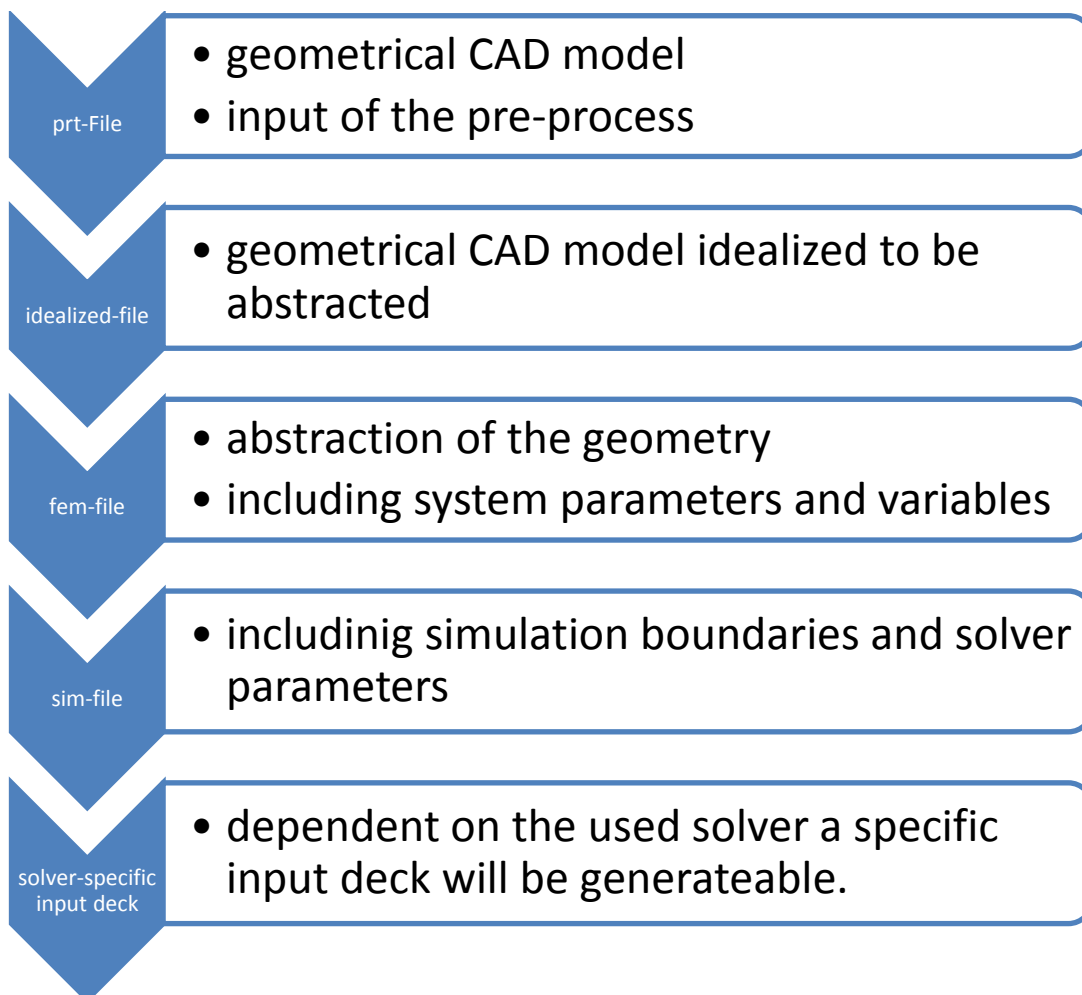


Figure 5-3 Basic Pre-Process-File-System of NX Advanced FEM

In Section 5.1.2, the solve-process in context of a basic simulation process will be discussed.

### 5.1.2 SOLVE-PROCESS

The solve-process needs fewer interactions with the user than the pre- or post-process. Usually, the input for the solver is the input deck. The input deck includes multiple files. In some cases, there is a direct transport of the information from the pre-process to the solver. In these cases, the input deck does not need to be generated. Most solvers document the solve-process in protocol files and store the results into result files. Some solvers also generate additional files such as files that can be used for a solve-process restart. The Figure 5-4 shows the files and data generated during the solve-process (Roensch, 2010)

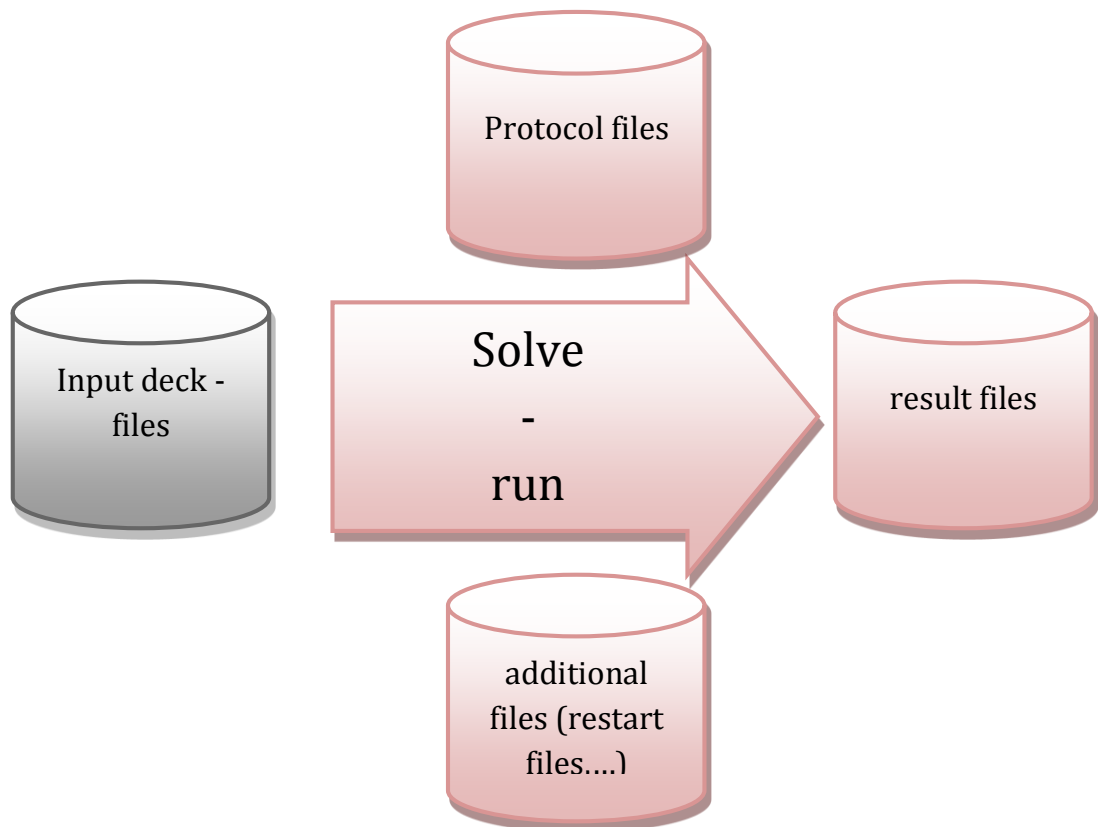


Figure 5-4 Files in the Solve-Process

If we use NX Nastran as an FEM solver, this solver handles the files and data described in the previously presented solve-process. In Table 5-1, the main files and data generated during the solve-process of the NX Nastran solver are listed (Siemens Product Lifecycle Management Software Inc., 2011).

Table 5-1 Main NX Nastran Files

File Type	File Suffix
Input deck	DAT
Protocol files	F04 and F06 and LOG
Result files	OP2

The results generated by the solve-process are used in post-process which will be discussed in Section Post-Process.

### 5.1.3 POST-PROCESS

The result files generated during the solve-process include a large amount of information. Reading through these files is almost impossible. However, visualisation and mapping of the results onto the pre-process information and data greatly improves interpretation of the results. So, the post-process visualises the results (Roensch, 2010). The result files are visualised in relation to the abstracted data and information of the pre-process. The results are in either two-dimensional or three-dimensional form and in coloured- mode data format. A graphical presentation is an efficient way to make the results much easier to understand and provide a means of interpretation in a reviewable format. This process is shown in Figure 5-5. For the most part, the interpretation of the results will also be documented. Such documentation i.e., during and after the result interpretation belongs to the post-process.

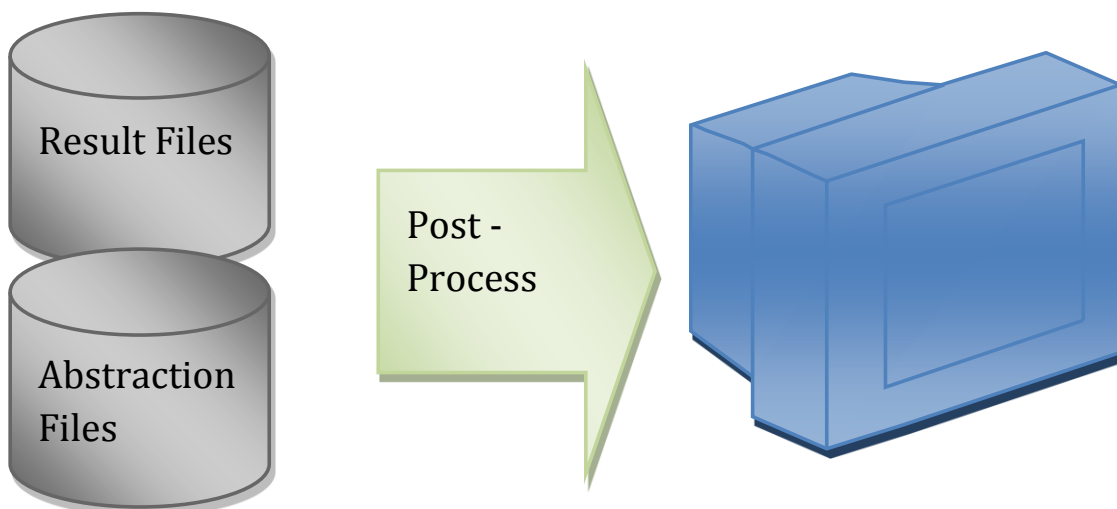


Figure 5-5 Post-Process Visualisation

The pre-, solve- and post-process constitute a basic simulation process. In the following examples, multiple simulations are combined. The represented basic simulation process should help to understand the following case studies and examples discussed in Sections 5.2 through 5.5.

## 5.2 CASE 1: CREATION OF A MOULDING-TOOL-SIMULATION-VISION FOR MOULDING-TOOL-OPTIMIZATION

SCHMIDT Gesellschaft für Werkzeug- und Formentechnik mbH is an engineering company specialising in the design of moulding tools. SCHMIDT Gesellschaft für Werkzeug- und Formentechnik mbH is under pressure from its competitors. So, the company undertakes to secure its place in the market by positioning itself as a high-end, high quality engineering business fully capable of offering a comprehensive, technically superior service encompassing the conception and design of moulding tools using optimised development processes. This can be achieved with simulation of the moulding tools. That will mean the simulation of the tool set and not mould-simulation. A benchmark should be employed to identify the ideal simulation tool.

The CAD models of the moulding tools will be provided as basis for the simulation by SCHMIDT. Thereby, the moulding tool set, as well the moulding process itself, should be addressed by the simulation. Additionally, the simulation should be reproducible by CAD-modelling-experts as opposed to simulation-experts.

Following this benchmark has established a unique relationship between SCHMIDT Gesellschaft für Werkzeug- und Formentechnik mbH and Siemens. In fact, greater knowledge about moulding tool simulations and their potential has been realised through this collaborative effort. The simulation work was carried out by me. Mahler & Schmidt(2012) analysed and evaluated the findings from this project. The materials in the paper will directly be integrated into the PHD thesis.

### 5.2.1 DISCUSSION OF MOULDING TOOL SIMULATION

Moulding tools are used extensively throughout the industry.

“Injection moulding is an ideal process for fabricating large numbers of geometrically complex parts. Many daily used items are injection moulded: mobile phone housings, automobile bumpers, television cabinets, compact discs and lunch boxes are all examples of injection moulded parts.” (Kennedy, 2008).

However, the moulding tools are expensive to develop and produce. In order to reduce costs, high volume production is necessary whilst retaining and guaranteeing quality of work. Therefore, in order to manufacture competitive plastic parts economically, the forming errors have to be reduced and the forming speed has to be increased (Clusterland Oberösterreich GmbH, 2012). This affects the forming process.

## I. Forming Process

“Injection moulding is a cyclic process. Initially, the mould is closed to form the cavity into which the material is injected. The screw then moves forward as a piston, forcing molten material ahead of it into the cavity.[ ...] When filling is complete, pressure is maintained on the melt and the packing phase begins. The purpose of the packing phase is to add further material to compensate for shrinkage of material as it cools in the cavity. At some time during packing, the gate freezes and the cavity is effectively isolated from the pressure applied by the melt in the barrel. This marks the beginning of the cooling phase in which the material continues to cool until the component has sufficient mechanical stiffness to be ejected from the mould. [...] When the moulded part is sufficiently solid, the mould opens and the part is ejected. The mould then closes and the cycle begins again.” (Kennedy, 2008).

This process is shown in Figure 5-6.

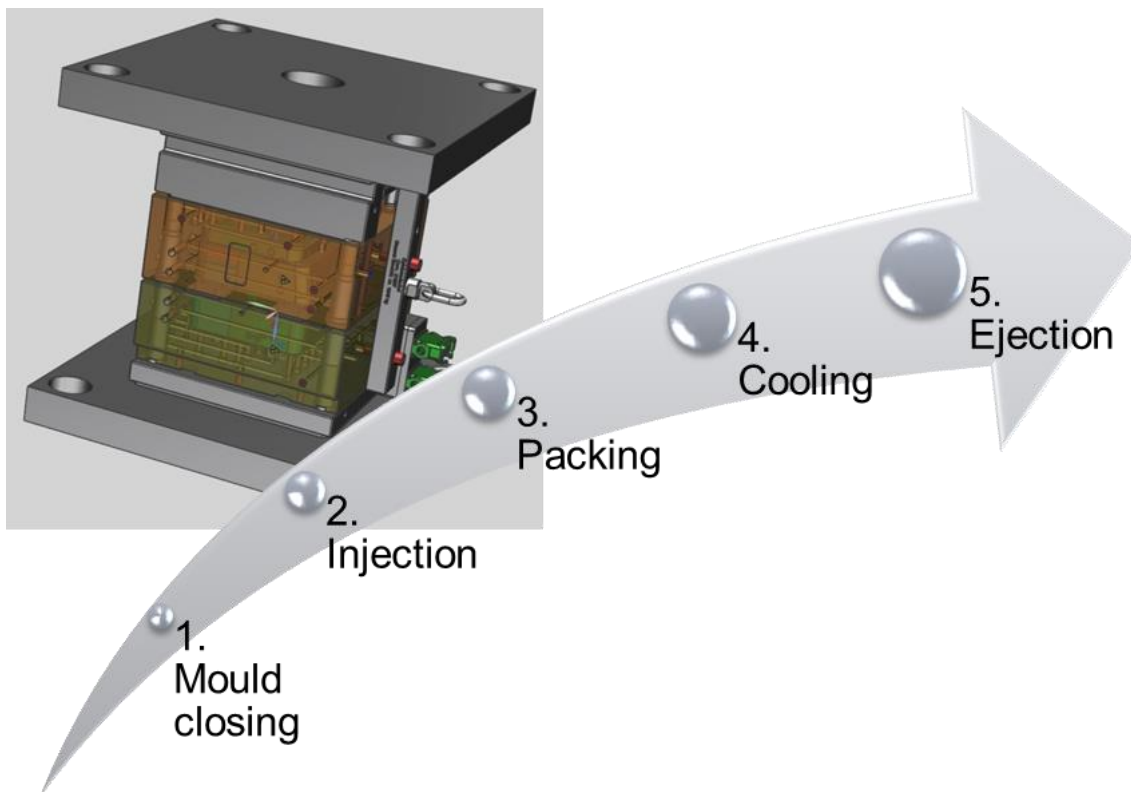


Figure 5-6 Case 1: Moulding Process



## II. Improvement of the Forming Process and Moulding Tool

The following are meaningful findings that can be used to increase the forming speed:

1. In the mould closing phase, the parts are moving. This causes extra load and results in stress and deformation of moulding tools and parts. In order to speed up the closing process, the mass of the parts of moulding tools and parts should be reduced and the drive force should be increased.
2. The injection temperature could be high since the injection material's property is improved. So the injection can be done with a higher injection velocity and higher pressure (WÜRTELE, LANGE & HUNGERKAMP, 2004). Similarly, the injection pressure and mould material temperature cause increased applied loads and results in stress and deformation of moulding tools and parts.
3. The pressure during the packing phase is increased (WÜRTELE, LANGE & HUNGERKAMP, 2004). This should reduce the errors due to material shrinkage. However, increased pressure causes higher applied loads and results in stress and deformation of moulding tools and parts.
4. The cooling process can be carried out more rapidly and in a controlled manner (Seidel, Brunner & Wißuwa, 2009) by using cooling pipes in the moulding tools. Unfortunately, rapid cooling causes higher applied loads and resultant stress and deformation of moulding tools and parts.
5. The ejection process can be improved in the same way as the mould closing process in Phase 1. The mass of the parts should be reduced and the drive force should be increased. However, mass movement causes higher applied loads and results in the stress and deformation of moulding tools and parts.

In summary, speeding up the process by reducing the mass of the moulding tools and increasing the applied loads on the moulding tools and parts result in increased forming errors and lower lifespan of the tools. A proper optimization of process parameters and tool sizes is necessary. This can be achieved with a coupled physics analysis, taking into account of all mechanical and thermal loads during a complete moulding cycle.

In Section 5.2.2, the simulation process and information flow to achieve such a simulation will be discussed.

### 5.2.2 DISCUSSION OF THE SIMULATION PROCESS AND INFORMATION FLOW IN CASE 1

The moulding tool simulation requires cooperation between the thermal, flow, structure and durability simulation disciplines. These four simulation disciplines can be integrated into a simulation in three phases:

1. Co-simulation of the thermal and flow simulation discipline.
  - a. The movement of the coolant is simulated with the flow simulation.
  - b. The temperature of the coolant and moulding tool is simulated with the thermal simulation.
  - c. The thermal and the flow simulation are directly coupled to a co-simulation to simulate the interaction of the moving coolant and the temperature of the coolant and the moulding tool.
2. Subsequently, the temperature results can be used to simulate the deformation and stress of the moulding tool caused by all applied mechanical and thermal loads.
3. The stress of the moulding tool causes damage. In a last step, the durability of the moulding tool is simulated based on previously generated stress results.

Incidentally, the source CAD models of this case study will include a high number of holes and blends. The influence of most holes and blends will be negligible for the simulation. Therefore the study will describe how to select which holes and blends should be ignored for the simulation thereby enhancing ease and performance. Additionally, the study will define how to select and blend a radius-limit-filter. It should be noted that only holes and blends in areas with low stress-peaks will be selected, whereas holes with force or boundary influences will not be selected and kept in the simulation. It has also to be observed that the lower moulding toolset will be fixed to the ground but the higher moulding toolset (the moving toolset) will be not fixed. Instead a closing force will be applied. By applying a closing force the simulation will resemble the behaviour of the moulding tool. The closing force influences the deformation and if the closing force is too low, the deformation of the moulding tool will be too high. In such case the gap between the higher and the lower moulding toolset could become too great and the work piece will not be produced to the required thickness tolerance. Such a case can be seen in Figure 5-8. Nevertheless, all these case example specific simulation influences, such as the hole and blend influences or the closing force influences, will have no impact on the PhD research project. The PhD research project will not concern itself with the simulation model details because the focus on the work and the work process will not be impacted using such a detailed viewpoint.

In the following, the three phases of the actual case study will be discussed.

## I. Discussion of Phase 1: Thermal and Flow Analysis

The simulation of the moving coolant is necessary to get the right cooling of the moulding tool caused by the coolant. So, it is required to couple the flow simulation of the coolant with the thermal simulation of the coolant which is achieved with NX CAE products such as NX Flow and NX Thermal (Ruel, 2011). The fluid volume of the coolant channels is derived from the solid model of the moulding tool (Siemens PLM Synchronous Technology Part2/2, 2008). Geometric models, the flow and the moulding tool are meshed. The coolant mesh has to be coupled to the moulding tool mesh for the thermal simulation, using NX surface to surface coupling (Inc., NX 8 Hilfe NX Nastran User Guide, 2011). Afterwards, the loads and boundaries are added. The applied loads, boundaries and solver parameters are:

- As load, the initial temperature of the injection mould is applied to the contact faces of the injection material.
- As boundary, the free convection of the moulding tool to the environment is defined.
- As load, the coolant is described between the inlet and outlet as recirculation loop with a predefinition of flow velocity and the heat exchanger parameters such as heat transfer coefficient, convection area and temperature. This achieves a coupling of thermal and flow 3D and 1D simulation. The meshed coolant volume is used for the 3D simulation. The two open ends of the meshed coolant volume are linked together by a mathematical formulation and represent the events between the two open ends. This will be possible by implementing a small 1D simulation solution in the used simulation authoring tool NX Advanced Simulation and NX Advanced Thermal.
- As solver parameters, a mixing length turbulence model has been chosen as the solvers mathematical handling of the numerical mathematic.
- As solver parameters, the simulation is done as a steady state simulation. This means that the temperature and flow simulation represent the constant state. The constant state appears when the losing and loading energy are both constant and similar to each other. So, the stress difference between the warmest and coldest temperatures in one production cycle process can be considered as negligible.

Figure 5-7 shows the result of this simulation case study.

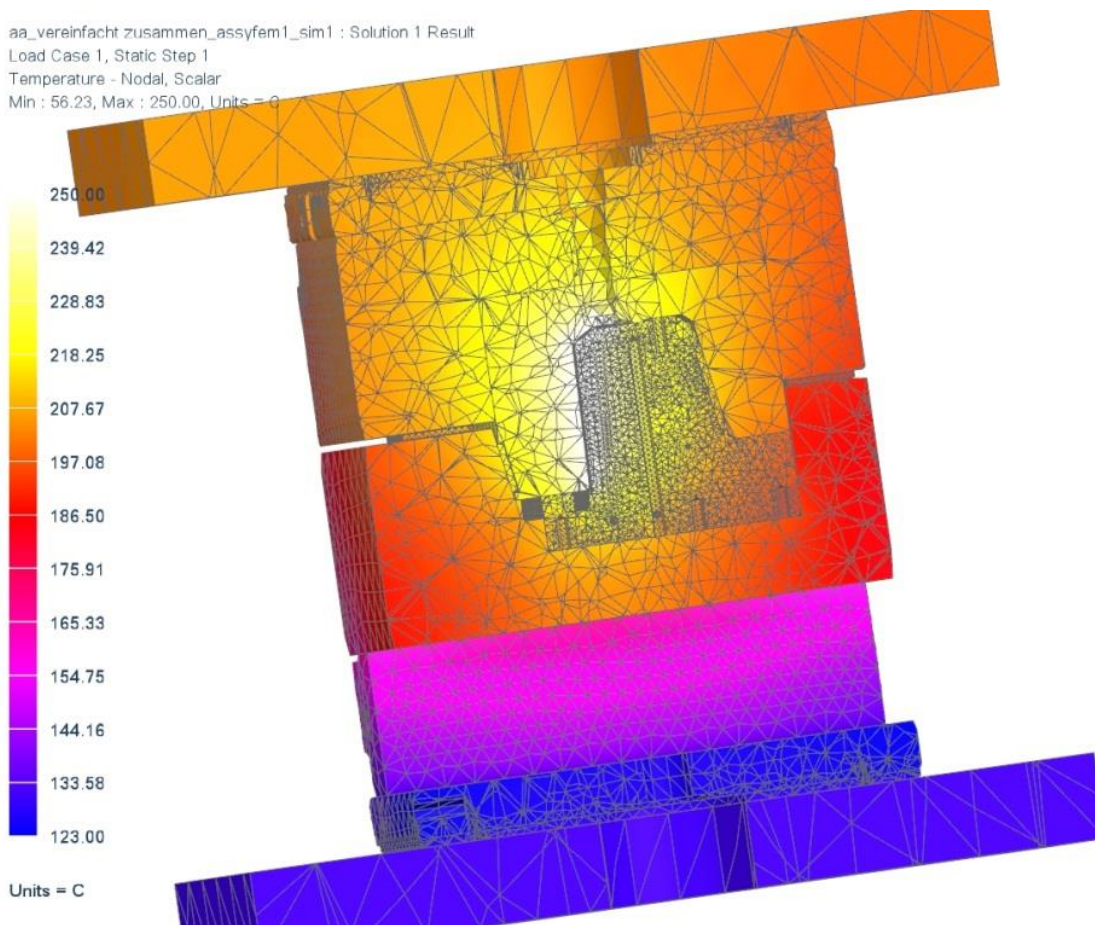


Figure 5-7 Result of Phase I Case 1

The thermal loads cause deformation and stress in the moulding tool. This deformation and stress will be simulated in Phase 2

## II. Discussion of Phase 2: Structure Analysis

For the structural analysis, only the solid mesh of the moulding tool is required. In order to generate the mesh of the moulding tool, the physical setup from Phase 1 is cloned. The clone is directly re-used to build the structural simulation. The mesh of the fluid volume and other data is deleted. On the mesh of the moulding tool (generated by cloning the solution of Phase I and editing the clone) the thermal distribution, generated in Phase 1, is applied as the load. The temperature field of the thermal analysis is used as an initial condition. In order to complete the physical setup of the structural analysis, bolt connections, pressure loads from the mould injection, closing forces and bolt preload forces, as well as the fixing of the ground plate are added. Contacts are also required. Contact face pairs are detected automatically. In most cases, the contacts are defined as surface-to-surface contacts with friction. Contacts are necessary to accurately measure the movement in the contact areas and the contact

pressures. The simulation is done with the linear solver NX Nastran SOL 101 (Inc., NX 8 Hilfe NX Nastran User Guide, 2011). Two sub cases have to be defined:

- Sub case 1 - to generate stress results of the warmed up moulding tool: Thermal loads representing the steady state thermal loading of the working moulding tool → once a day.
- Sub case 2 - to generate stress results of a lot production: Additional loads are required. These constitute the closing force and injection pressure caused by one production cycle → for each work piece.

A result of Sub case 2 is shown in Figure 5-8.

The deformation and stress results are focused in this phase. They help to identify critical design positions. Additionally, they help to decide if further analysis is required.

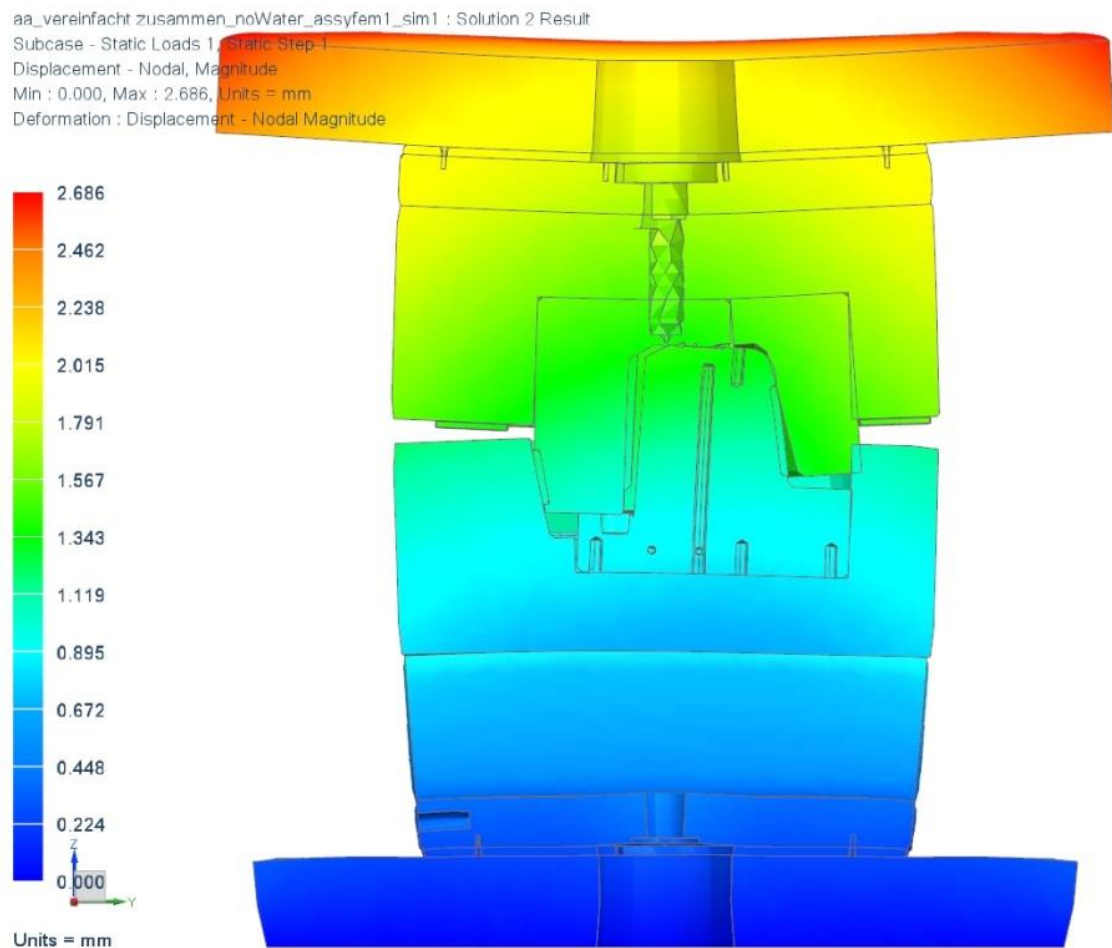


Figure 5-8 Result of Phase 2 Case 1

The stress results of the Sub case 1 and Sub case 2 cause damage to the moulding tools and parts material. This damage can be simulated. In Phase 3, this is done as a fatigue simulation.

### III. Discussion of Phase 3: Fatigue Simulation

The fatigue simulation is based on the stress results of Phase 2. A fatigue simulation requires a high quality mesh. So, NX Advanced Simulation was selected to provide an automatic mesh refinement algorithm taking into account local stress error deviations (Siemens Product Lifecycle Management Software Inc., 2011). The mesh refinement also causes a large number of data and file sizes, and hence requires lengthy time and resource investments. The fatigue simulation does not require all the parts used in Phase 2 - only one part is necessary. A reduced simulation model can be generated by cloning and editing the simulation of Phase 2. This clone includes only the required part for the durability simulation. The simulated deformation (Sub case 1 and 2) of this part, completed in Phase 2, can be applied locally as a constraint. The refinement of the mesh and the deformation loads generates the stress results with the required high quality for the fatigue simulation. The fatigue should represent one production day. So, two load cases are summarised:

- One cycle of Sub case 1 that represents the fatigue damage of warming the moulding tool and
- {Number of produced work pieces a day} cycles of the Sub case 2 that represent the damage caused by the loads of one production lot.

The fatigue results should help to optimise the design of the moulding tool. In order to optimise the moulding tool, only relative and non-absolute fatigue results are necessary. However, it is common practice to compare relative fatigue results in order to compare different design variants. This practice is used to improve the moulding tool.

The result of the fatigue simulation is shown in Figure 5-9.



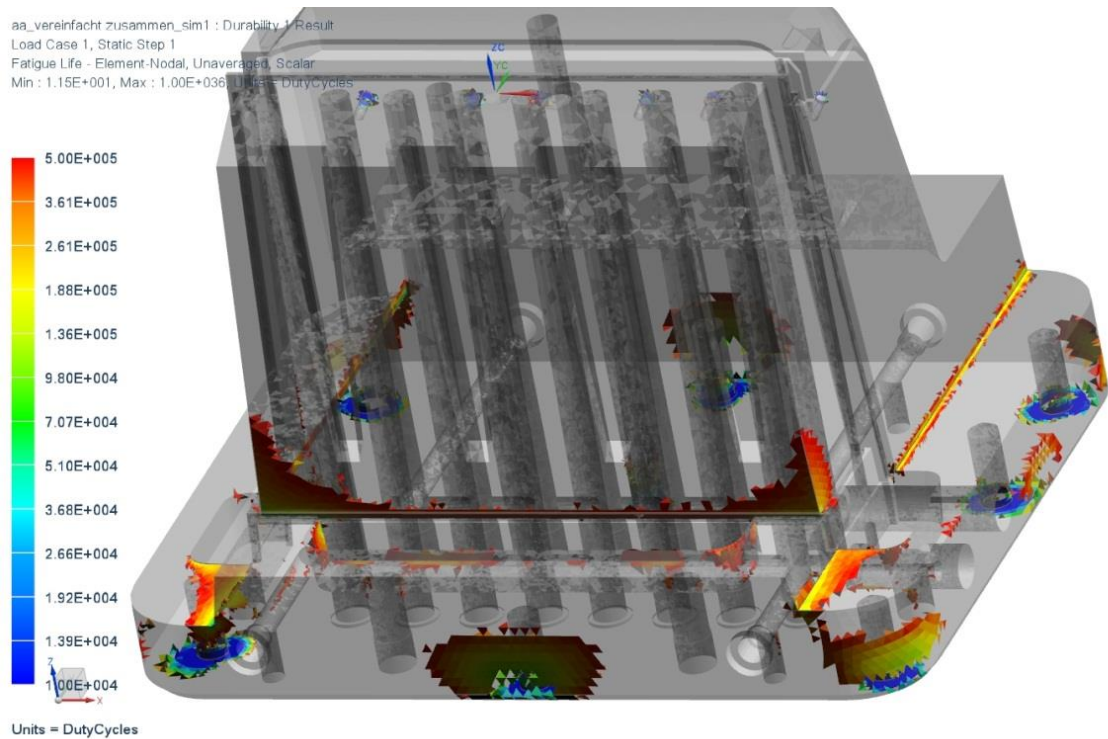


Figure 5-9 the Result of Phase 3 Case 1

The simulation process during Phase 1 to Phase 3 uses and generates multiple files, data and information. The handling of these files, data and information plays an important role to realise this simulation process. This will be discussed in Section 5.2.3.

### 5.2.3 DISCUSSION ABOUT THE USED AND GENERATED FILES, DATA AND INFORMATION OF CASE 1

In this section, the used and generated files, data and information will be discussed and described. This will be carried out for each phase of the simulation process.

#### I. Phase 1: Thermal and Flow Analysis

The source of the simulation is CAD geometry models stored in multiple files. These files are organised to represent the hierarchical order as assemblies and parts. NX is the used CAD and CAE tool. NX provides the possibility to generate idealized parts directly in the CAE system. The behaviour of this file should be to generate geometry that is idealized for meshing. For each CAD part, if it is required in the simulation, an idealized part is generated.



The meshing is done in the next step. In NX, the meshing is done in an extracted file, called FEM file. The FEM files are dependent upon the idealized parts. The FEM files mesh the individual idealized parts, whereby, the modelled idealised parts are based on individual CAD parts. The moulding tool is an assembly. So, an assembly mesh called AFM file is generated. The AFM file is dependent on the assembled CAD model. In the AFM file, the FEM files are sampled hierarchically, similar to the CAD assembly and dependent upon the CAD model that includes the assembly information. Using the information from the dependent CAD model (assembly), the AFM mesh knows how to position the individual FEM files in a three-dimensional form. The original assembly of the moulding tool combines 134 CAD part files. The simulation of the moulding tool requires only 15 CAD parts, and the top assembly. For each of these 15 parts, an idealised part file and an FEM file are generated. The idealised parts are also used to generate the cooling pipes volume which means that the parts are able to include multiple geometrics such as cooling pipe volume and moulding tool volume. Also, the FEM file can include multiple meshes. Subsequently, an AFM-file is generated dependent upon the CAD assembly part. This AFM file organises the 15 FEM files associated with the CAD assembly. Dependent upon the AFM file, a SIM file is generated. The SIM file is used to apply loads and constraints, and solver parameters. The described file structure is shown in Figure 5-10. The simulation for the coupling of the thermal and flow simulation for Phase1 is achievable using only one SIM file and one solver.

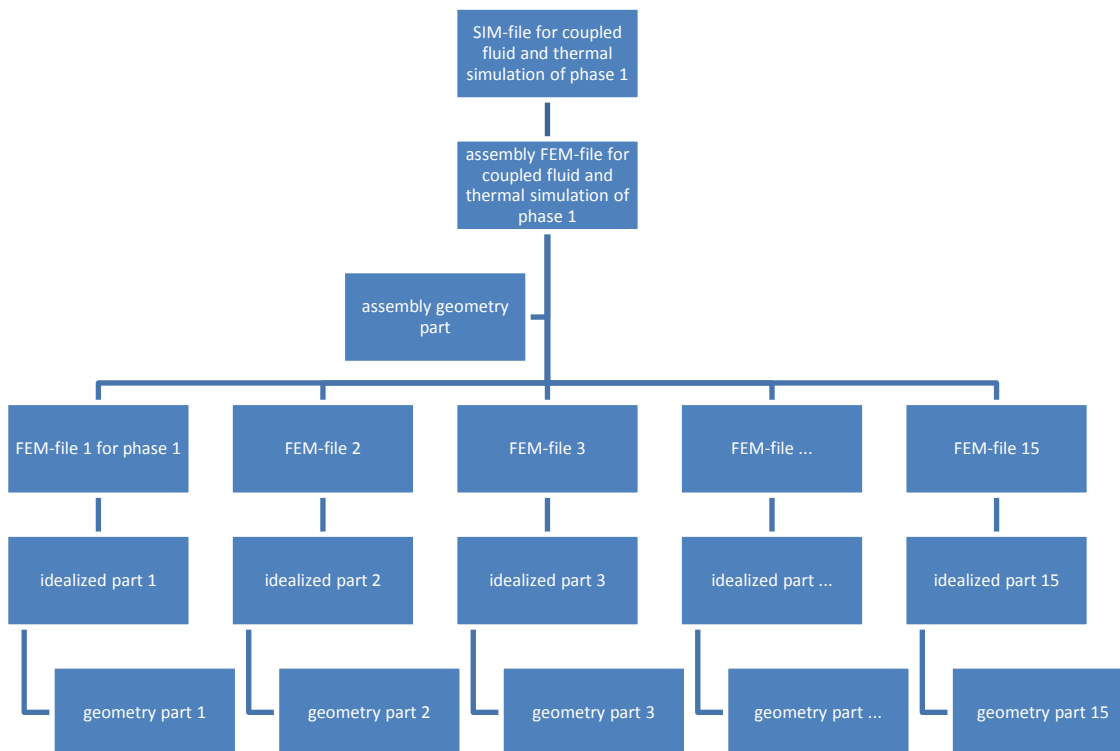


Figure 5-10 Phase 1 Simulation Structure of Case 1

The simulation of Phase 1 is achieved with NX Advanced Simulation as the pre-processor and NX Thermal and NX Flow as solvers. The pre-process of Phase 1 produces an input deck file

readable by the coupled NX Thermal and NX Flow solvers. The solve-process generates results and protocol files. NX Advanced Simulation, used for the pre-process, is also used for the post-process of the results. The important result is the thermal distribution in the moulding tool. For the re-use of this result, the thermal distribution is extracted and stored with the NX Advanced Simulation as a single file (ASCII format). The thermal distribution in this file is used as input for the simulation of Phase 2 and Phase 3.

## II. Phase 2: Structure Analysis

In order to generate stress and deformation results, a structure simulation is carried out in Phase 2. This simulation requires the same structure as was generated in Phase 1. In additionally, most files and data from Phase 1 can be re-used. Changes are required in the SIM file, AFM file and one FEM file. In order to re-use the data and files, NX Advanced Simulation provides clone technology so these three files are cloned and saved under new names. By using this clone functionality in addition to file cloning, the hierarchical dependencies between existing files from Phase 1 are assumed. In Figure 5-11 the structure of Phase 2 is shown.

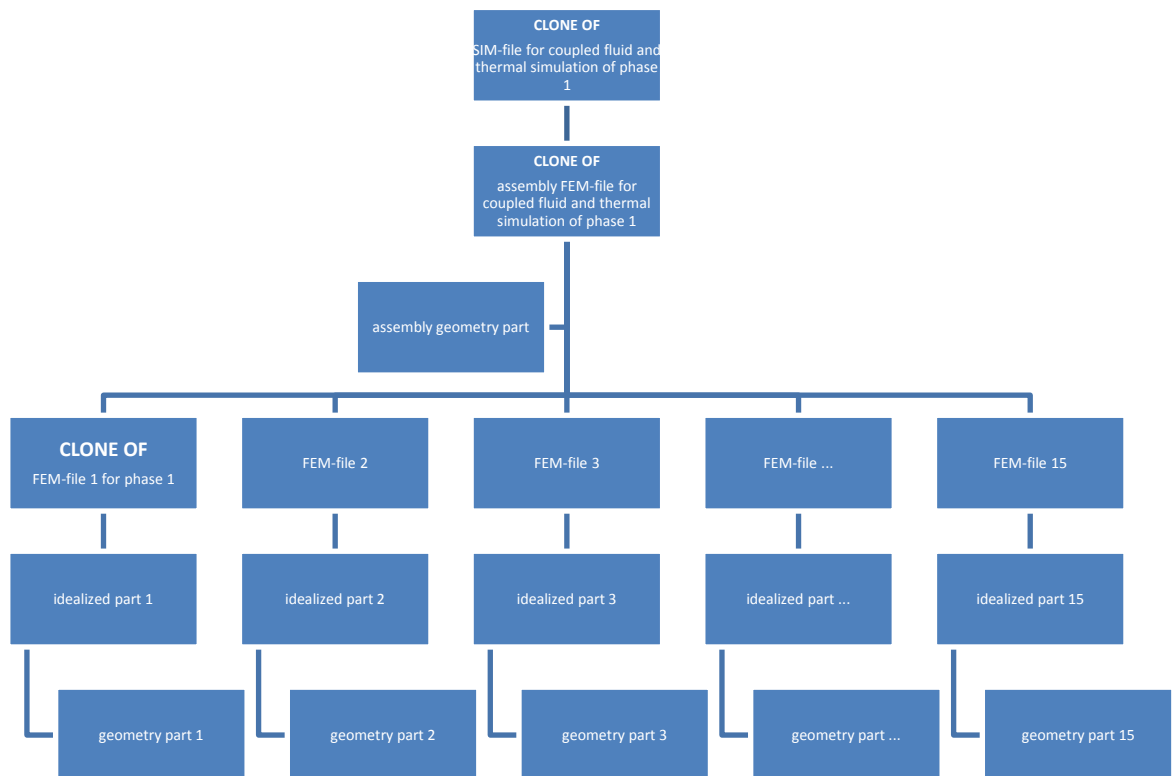


Figure 5-11 Phase 2 Simulation Structure of Case 1

As mentioned above, there are changes in the three cloned files. These changes are:

- In the cloned FEM file 1, the coolant meshes are deleted.
- In the cloned assembly AFM file, the FEM file1 is replaced by the cloned FEM file 1. The other FEM-files are the originals from the original AFM file. The replacement of the cloned FEM file 1 affects the coolant meshes which are included in the structure simulation of Phase 2.
- In the cloned SIM file, the AFM file is replaced by the cloned AFM file. Additionally, the predefined simulation definitions of Phase 1 is deleted and replaced by a structure simulation definition (see also Phase 2: Structure Analysis).

An unusual case is the thermal load case defined in the cloned SIM file. The thermal distribution is based on an additional file. This file is generated in the post-process of Phase 1. Here, the thermal distribution is extracted from the results of Phase 1 and saved in a specialised file. This is shown in Figure 5-12.

The structure simulation of Phase 2 simulates the stress and deformation results of the moulding tool assembly. Two different simulation sub cases (see also Section 5.2.2 II) are generated. Both of them generate an input deck for the solver, the results and log files during the solve-process. Based on the resultant data, the staff responsible for this can decide if a fatigue simulation is required. Post-processing in Phase 2 can be used to identify additional local constraints and loads for the fatigue analysis which is stored in a specific file, similar to that carried out in Phase 1. The used and generated files of the fatigue simulation in Phase 3 will be discussed next in Phase 3.

### III. Phase 3: Fatigue Simulation

In Phase 3, the fatigue simulation is focused on one or two critical moulding tool parts, specifically, this is the CAD Part File 1. So, the existing FEM file 1 is dependent on the Part File 1 and will be cloned. In order to achieve a high mesh and result quality, the mesh is automatically refined. The refinement of the mesh is done in a loop until a predefined target is achieved. During the adaptive mesh refinement, control files are written which are deleted at the end of the adaptive re-meshing. The solver result, protocol files and the cloned FEM file 1 are overwritten. This planned refinement means that a direct use of the original FEM file 1 can change the FEM file 1 used in the simulation of Phases 1 and 2 (see 5.2.2 III). So, a cloning of the FEM file 1 is required for Phase 3. Based on the cloned FEM file 1, a SIM file is generated. Similar load cases, boundary conditions or solver parameters of the simulation (SIM-file) in Phase 2 are applied. Furthermore, the thermal load case with the additional thermal distribution file is applied. The derived constraints of the simulation results of Phase 2 can be a forced deformation of local points, such as holes. The deformation dimension is identified by the post-process of Phase 2. The solver parameters are similar to the structure

simulation of Phase 2 using the NX Nastran SOL101 solver. The results show the deformation and stress for the simulation model in Phase 3 but with a higher quality than in Phase 2.

Refined mesh with a better result quality is used for a fatigue simulation. Two structure simulation sub cases are required to summarise the fatigue of two different types of fatigue damages. Both of the structure simulation sub cases generate an input deck for the solver, the results and log files during the solve-process. Based on the result files and the fatigue solver parameters, the fatigue solver generates results and log files. The fatigue simulation definition is also realised in the SIM file and with NX.

These files, data and information of the moulding tool simulation should be managed by TEAMCENTER. In Section 5.2.4, the data management with TEAMCENTER for this case example will be discussed.

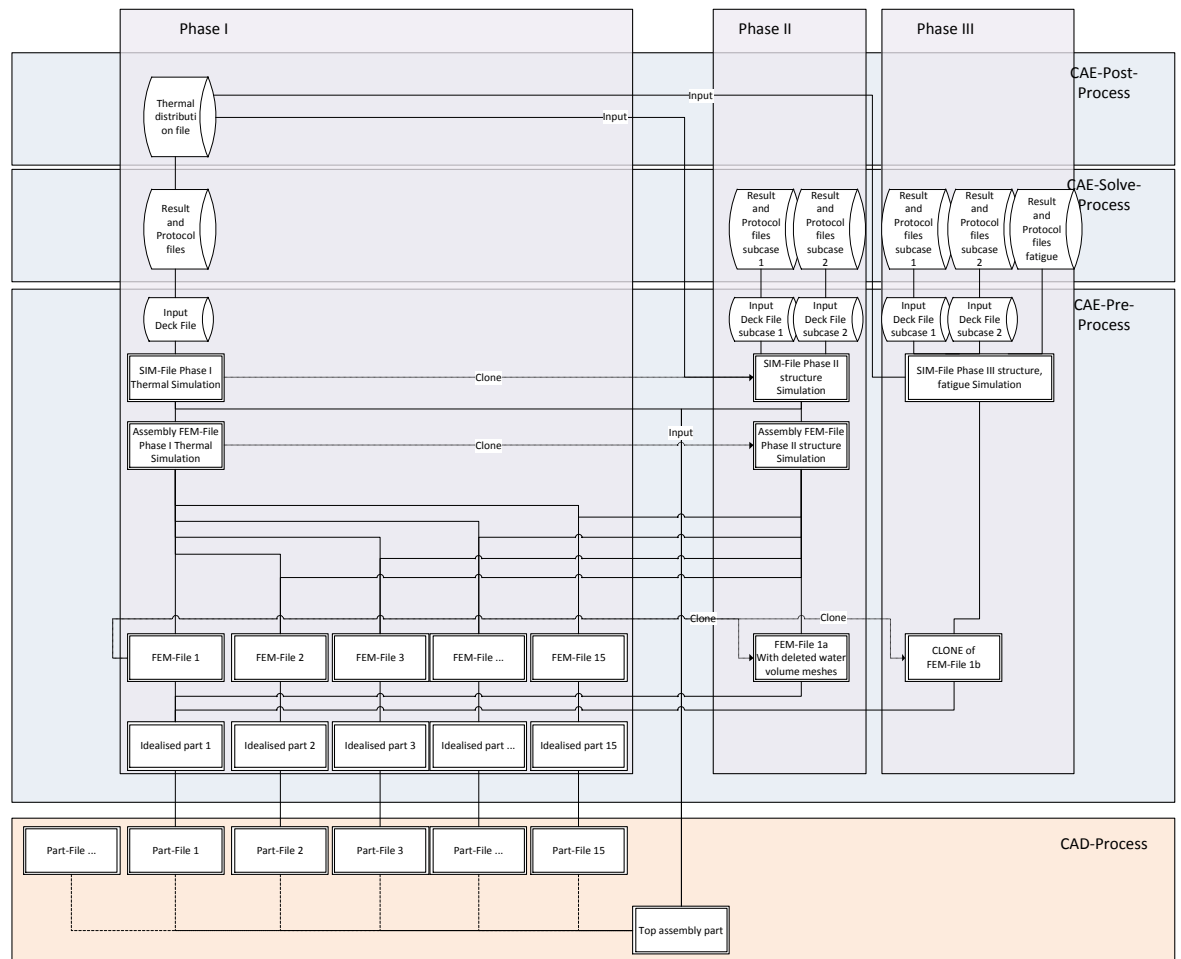


Figure 5-12 Simulation Structure of Case 1

#### 5.2.4 DATA MANAGEMENT OF CASE 1 WITH TEAMCENTER

The case study includes an integration of the system-simulation into the PLM framework TEAMCENTER. The management of all the required files and the support of the simulation process should be achieved.

In Phase 1, multiple CAD models are managed as CAD item Revisions. 15 idealized parts (to generate the ideal geometry for meshing) are generated on 15 CAD item Revisions. Each idealized part is stored as dataset under a new CAEGeometry item Revision. A CAESource and CAETarget relationship is used to link each of the 15 new CAEGeometry item Revisions to the responding CAD item Revision source (Siemens Product Lifecycle Management Software Inc, 2012). Each individually idealised part is the source for a FEM file (to generate the mesh of the geometry). The 15 FEM files are stored as dataset under 15 new CAEModel item Revisions. In order to represent the source dependencies of a FEM file, a CAESource relationship points from each CAEModel- item Revision to the source CAEGeometry item Revision. Additionally, to define the represented original geometry of the FEM file, a CAETarget relationship points from each CAEModel item Revision to a CAD item Revision (Siemens Product Lifecycle Management Software Inc, 2012). In order to combine the 15 FEM parts, they are subordinated by an AFM part. This AFM part is dependent on the assembled CAD model including the underlying CAD models. The AFM part file is stored as a dataset under a new CAEModel item Revision. In order to manage the dependency to the assembled CAD model, this CAEModel item Revision is linked to a CAE Source and a CAETarget relationship pointing to the CAD item Revision with the assembled CAD model dataset. The hierarchical structure of the 15 CAEModel item Revisions with the single FEM parts to the new CAEModel item Revision with the AFM part are managed with BOM view technology (Siemens Product Lifecycle Management Software Inc., 2012). The BOM view of the CAEModel item Revision with the assembled FEM part ordiates the 15 individual CAEModel item Revisions. The thermal simulation is stored as SIM file (called thermal SIM file). The thermal SIM file is stored as a dataset under a new CAEAnalysis item revision. The source of the thermal SIM file is the assembled FEM part. This is represented by a CAEDefining relationship pointing from the CAEAnalysis item Revision to the CAEModel item Revision with the AFM part. Based on the thermal SIM file, a solve run of the simulation is executed. The result files, generated by the solve run, are stored as datasets under the CAEAnalysis Revision. With post-processing, an additional file should be generated. This file should include the description of the thermal distribution and should be importable for later simulation steps. The extracted file is managed as a dataset under the CAEAnalysis item revision of the thermal SIM file. The items, as well the relationships and the datasets, are shown in Figure 5-13.

In Phase 2, the CAEModel-Item Revision 1 including the FEM file 1 is cloned. The FEM file 1 includes the mesh of the geometrical CAD model part. However for the next simulation, the mesh has to be changed. In order to keep the existing simulation model constant and have a changeable item and dataset, the CADModel item revision 1 has to be cloned. A new

CADModel item revision and dataset including a copy of the FEM file 1 is generated as a result of the cloning. This new item is the CADModel item revision 1a and the underlying dataset FEM file 1a. The source of the new generated CADModel item revision 1a is the same as the original CADModel item revision source. This dependency is represented by a new CAE Source relation from the new CAEModel- item revision 1a to the CAEGeometry item revision 1. The represented CAD item revision of the new CAEModel item revision 1a is also the same as in the original CAEModel item revision 1. This dependency is represented by a new CAE Target relationship from the CAEModel item revision 1a to the CAEGeometry item revision 1. The changes of the mesh can now be completed.

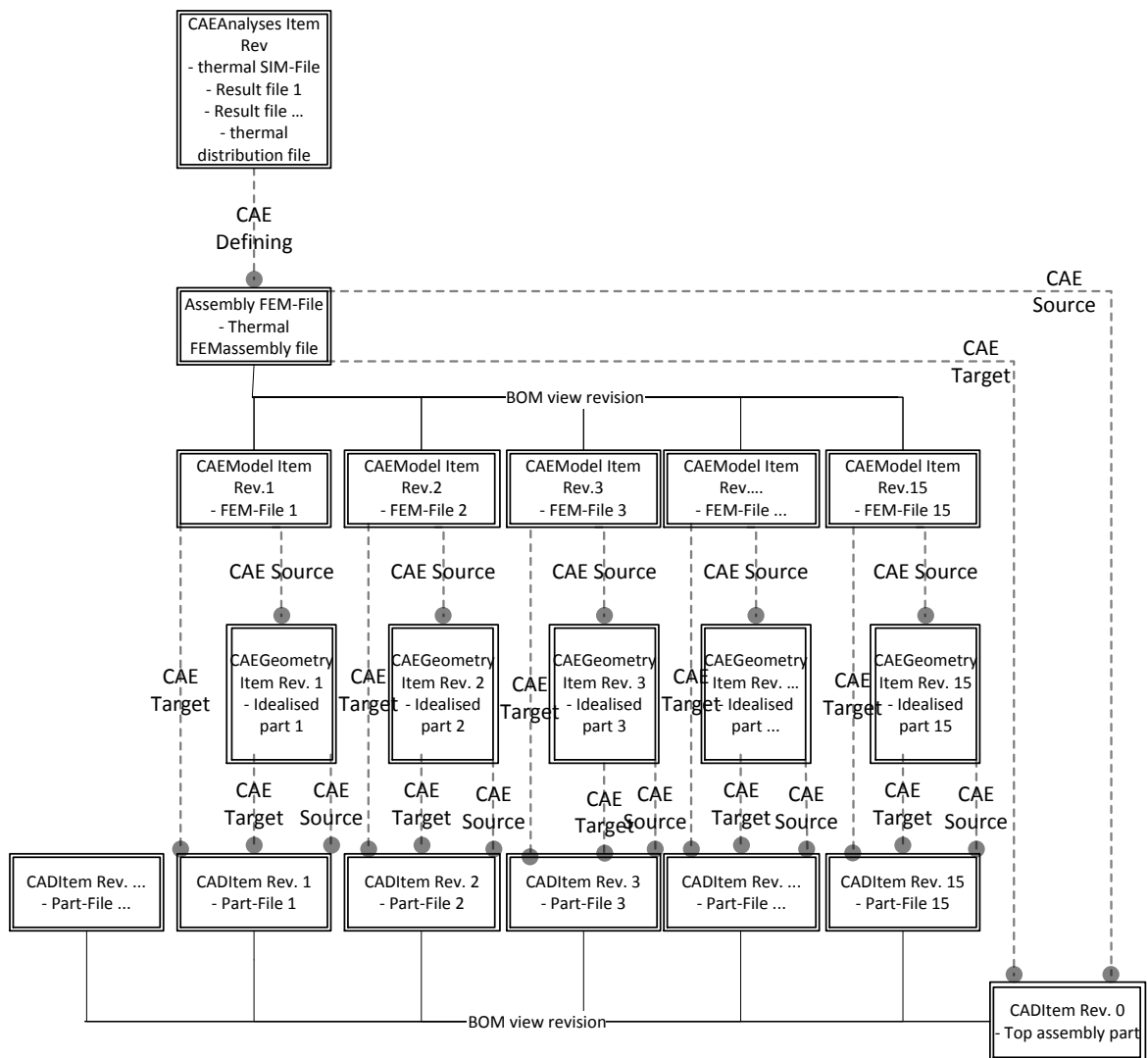


Figure 5-13 Metadata Structure of Case 1 Phase 1

In Phase 2, the AFM file of Phase 1 also needs to be cloned. The cloning and the behaviour of the cloning are similar to the cloning of the FEM file. A new CAEModel item revision is generated, called CAEModel item revision 0a and a copy of the dataset (thermal FEM assembly file) is generated, the FEMassembly file. The assembly mesh of the CAEModel item revision 0a requires the new generated part mesh of the CAEModel item revision 1a instead of the part mesh of the CAEModel item revision 1. The hierarchical ordering is changed and leads to the change of the BOM view of the CAEModel item revision 0a.

Based on the data of the CAEModel item revision 0a, the structured SIM file is built. This SIM file is stored under a new CAEAnalysis item revision as a dataset and the CAEDefining relationship is pointing to the CAEModel item revision 0a. The SIM file generation requires the thermal distribution file generated in Phase 1 as input. The thermal distribution file is managed as a dataset under the CAEAnalysis item revision of Phase 1. This dependency is represented by a CAEInclude relationship pointing from the CAEAnalysis item revision of Phase 2 to the CAE Analysis item revision of the Phase 1.

However, the CAEInclude relationship does not point to the dataset used as input for Phase 2 simulation and for this reason, this relationship is not precise. Additional information to identify the precise dataset and file in the CAEAnalysis item revision of the Phase 1 is - required. So, an out-of-the-box solution is not available. Customization would be useful. Additionally, the support of the CAEInclude relationship is not given in the case of the simulation authoring tool NX Advanced Simulation. In this case, the thermal distribution file dataset of the CAEAnalysis Item Revision in Phase 1 is exported manually.

The result files generated by the solve run of Phase 2 is stored in datasets under the CAEAnalysis item revision. The post-process of the result files does not produce additional files.

The items and the relationships and the datasets are shown in Figure 5-14.



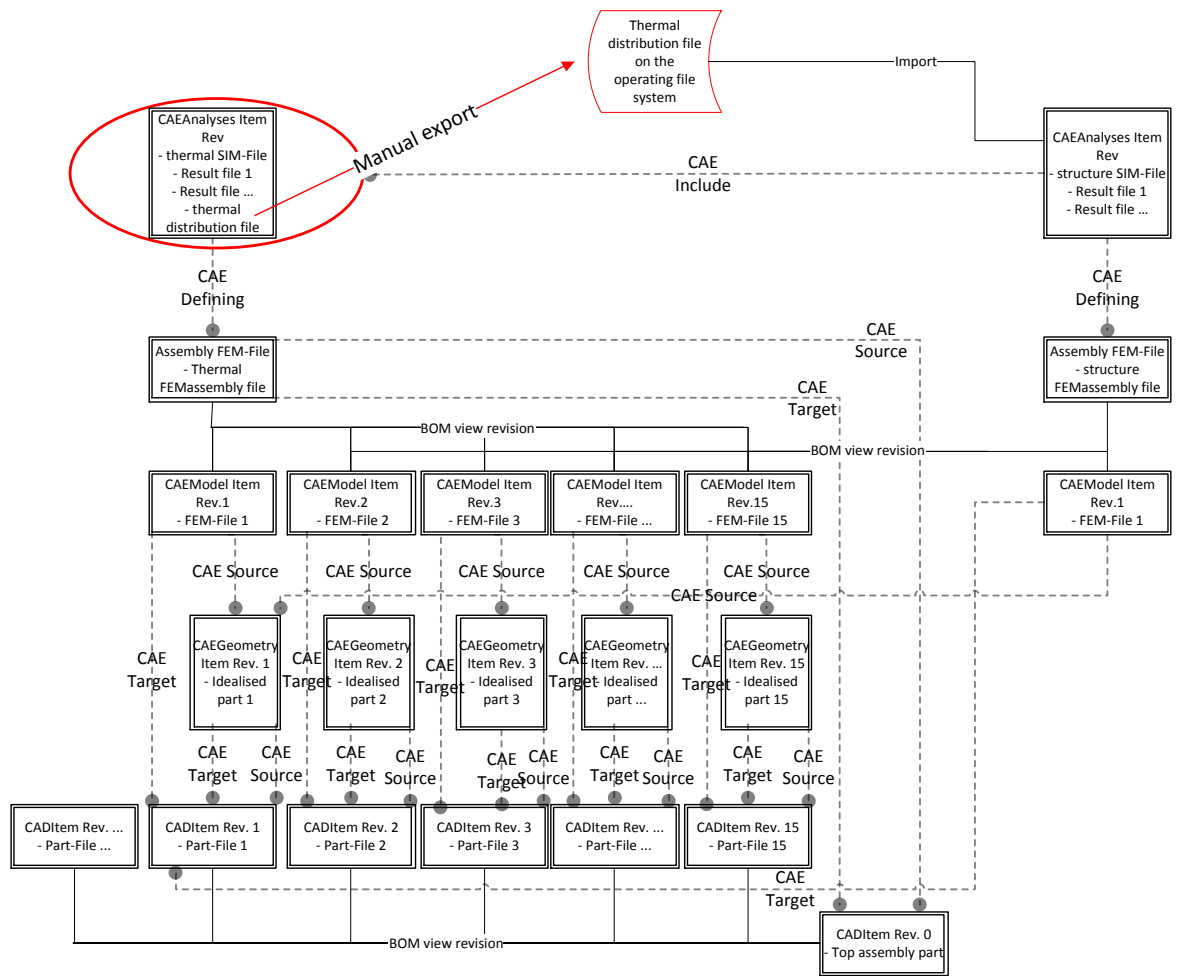


Figure 5-14 Metadata Structure of Case 1 Phase 1 and 2

In Phase 3, the CAEModel item revision 1 is an additional cloning that causes the generation of a new CADModel item revision, a copy of the datasets called CADModel item revision 1b and dataset FEM file 1b. Similar to the cloning in Phase 2, the new generated CADModel item revision 1b is dependent on the CAEGeometry item revision 1 represented by a new CAESource relationship and the CAD item revision 1 represented by a new CAETarget relationship. Based on the data of the CADModel item revision 1b, a detailed structure SIM file is created. This file is managed as a dataset under a new CAEAnalysis item revision and a CAEDefining relationship points from the CAEAnalysis item revision to the CADModel item revision 1b to represent the source dependency. The detailed structured SIM file generation requires the thermal distribution file generated in Phase 1 as input. The thermal distribution file is managed as a dataset under the CAEAnalysis item revision of Phase 1. This dependency is represented by a CAEInclude relationship pointing from the CAEAnalysis item revision of Phase 2 to the CAE Analysis item revision of the Phase 2.

The re-use of the thermal derivation file generated in Phase 1 as input for Phase 3 is similar to the re-use of the thermal derivation file as input in Phase 2. This means that the relation to the thermal derivation file is not precise enough and requires additional information. As discussed in Phase 2, the solution is that the thermal distribution file dataset of the CAEAnalysis item revision in Phase 1 are exported manually.

In Phase 3, three simulation sub cases are created. The first sub case should generate stress and deformation results caused by the thermal and pressure loads. Based on the first simulation sub case, the second simulation sub case should automatically improve the mesh quality. The mesh is reworked automatically. The improvement of the mesh also improves the quality of results from Sub case 1. The simulation Sub case 2 runs the mesh refinement and simulation from Sub case 1 in a loop process. After achieving a predefined mesh quality, an additional simulation sub case is created based on simulation Sub case 1. The third simulation sub case should generate durability results. The results files and data of the three simulation sub cases are stored in datasets under the CAEAnalysis item revision of Phase 3. The post-process of the results do not produce additional files.

The items, relationships and datasets are shown in Figure 5-15.

In summary, TEAMCENTER provides good support and management of individual simulation models. Traceability of the individual simulation model and its sources and represented product parts are achieved. Nevertheless, TEAMCENTER does not support multi-disciplinary and inter-disciplinary interaction of individual simulation models. The unique traceability of multi-disciplinary and inter-disciplinary interaction will not be achieved and the required data will neither be provided nor identifiable. Furthermore, the unique review of the multi-disciplinary and inter-disciplinary interaction between the individual simulation models with TEAMCENTER will not be achieved.

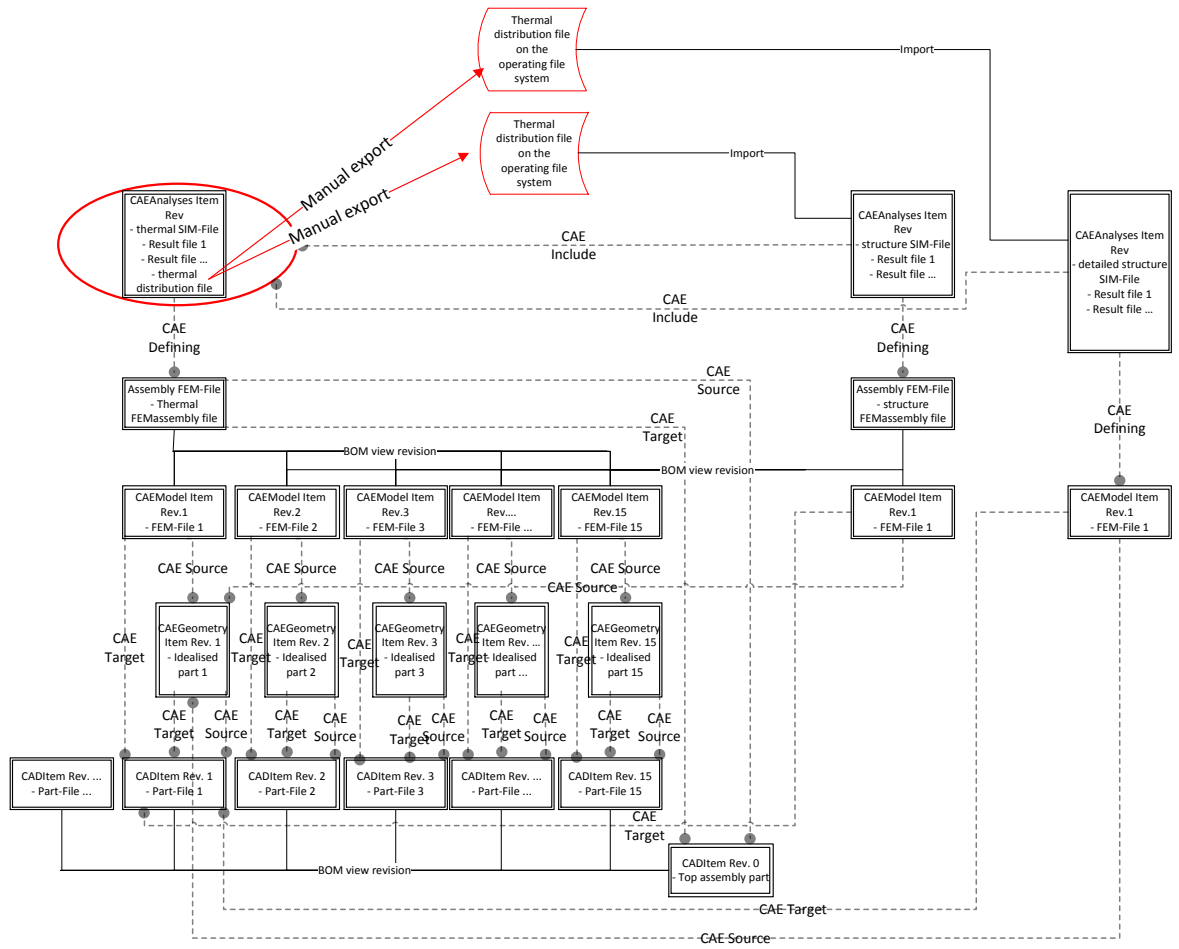


Figure 5-15 Metadata Structure of Case 1 Phase 1, 2 and 3

In the next case, an improvement in the development process of extrusion section production tools is achieved by the interaction of simulation models and design models. This will be discussed in Section 5.3.

### 5.3 CASE 2: IMPROVED DEVELOPMENT PROCESS OF EXTRUSION SECTION PRODUCTION TOOLS BY SIMULATION AND DESIGN MODEL INTERACTION

The customer of this project is a producer of semi-finished metal products. Some of these semi-finished products are produced as extruded sections. The extruded section is produced by an extruding machine including a die.

The extruded sections should comply with the requirements of the customers. An optimization of the disposed extruded section can be achieved by optimizing the die. The

company searched for a solution to achieve the die optimization. The use of simulation was also investigated.

Siemens Industry Software provides simulation applications that are ideal for a semi-finished goods producer. The customer was interested in benchmarking the possibilities with tools from Siemens Industry Software. My responsibility, from a Siemens Industry Software perspective, was to take care of the use case implementation and I determined an optimization of the development process by combining the simulation with the design applications in NX. In order to understand the data management issues associated with this case study, the extrusion section production process has to be presented at the outset. This will be discussed in Section 5.3.1.

### 5.3.1 DISCUSSION OF EXTRUSION SECTION PRODUCTION PROCESS

As discussed above, the main concern is to optimise the extruded section. This requires the optimization of the die tool. So, the extrusion section production environment and the working environment of the die have to be integrated into the development process. Temperature differences have a considerable influence on the die and extrusion section production process. The manufacturing environment temperature of the die is around 20° Celsius but the temperature of the extrusion section production process is much higher. The extruded section variation to the expected form is also influenced by this temperature discrepancy. It is assumed that it is possible to simulate the geometrical characteristics of the die at the extrusion section production process and the die manufacturing process. In this case, geometrical characteristics dictate the geometrical deformation of the die. The extrusion section is influenced by the big difference of the temperature discrepancies. The extrusion section should achieve an expected form at room temperature but the extrusion section production process is at a much higher temperature. In the following, the extrusion section production process will be explained in more detail.

In the extrusion section production process, which is the working phase of the die, a

“material is pushed or drawn through a die of the desired cross-section. The two main advantages of this process over other manufacturing processes are its ability to create very complex cross-sections and work materials that are brittle, because the material only encounters compressive and shear stress. It also forms finished parts with an excellent surface finish.” (WIKIPEDIA, 2012)

Udomphol & Technology (2012) explains: “Most metals are hot extruded due to large amount of forces required in extrusion.” The use case of the semi-finished goods producer optimises the development process of the die used for a hot extrusion.

In the extrusion process, the machine is loaded with hot material. A punch forces the hot material “to flow in the same direction as the punch (Udomphol & Technology, 2012).” The flowing hot material has to pass through the die tool. The die tool forces the hot material to change the form. This process is shown in Figure 5-16 and Figure 5-17.

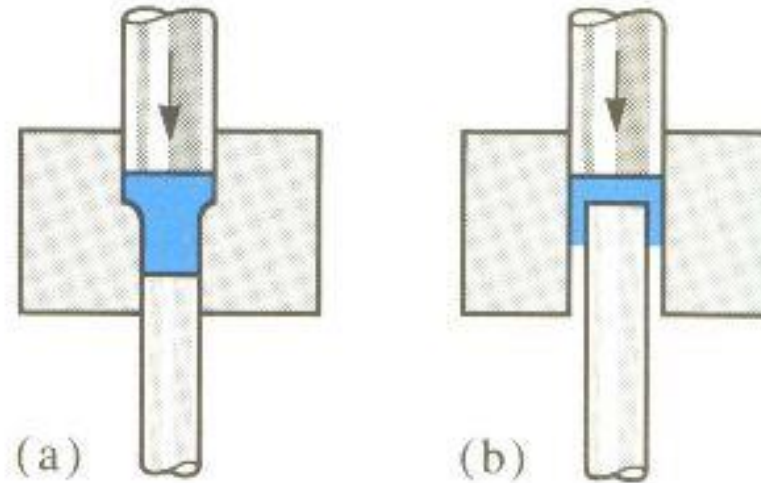


Figure 5-16 Extrusion Process (Udomphol & Technology, 2012)

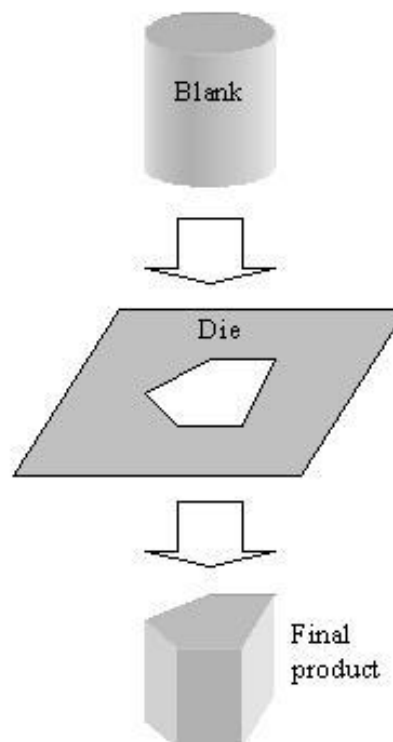


Figure 5-17 Extrusion Process (WIKIPEDIA, 2012)

The product coming out of the extrusion machine is the hot extrusion section. However, the end-product demanded by customers of the semi-finished goods producer, is the cold extrusion section. Therefore, the customers define the exact design and tolerances of the cold extrusion section. Besides, the hot production process of the extrusion section causes a difference between the expected and produced geometrical characteristics of the extrusion section. The geometrical material form changes during temperature changes. Therefore an improvement in the die development process can be optimised to reduce the differences between expected and produced semi-finished goods. Section 5.3.2 will discuss how a simulation process and an information flow will improve the development process.

### *5.3.2 DISCUSSION OF THE SIMULATION PROCESS AND INFORMATION FLOW IN CASE 2*

Simulations can help to develop the hot extrusion section and expected extrusion section at room temperature, as required by the customer. Simulations can also be used to design a cold die tool produced at room temperature. So, the simulation calculates the deformation difference between the cold extrusion section (initial temperature is room temperature) and the hot extrusion section (load temperature is production process temperature of the material). This deformation can be used to demonstrate the difference between the geometrical description (available as a CAD model) of the cold extrusion section and the geometrical form of the hot extrusion section.

The die shapes the hot material into the geometrical form of the hot extrusion section. So, the hot extrusion section geometric is included at the end of the die as a negative form. In the case of the production process, the die and the extrusion section are in a hot state, required for the production process. The temperature distribution of the die volume is inconsistent. On the contrary, the hot material and the fresh extruded hot extrusion section can be seen as loaded with an even and constant temperature. The die is warmed up on the inlet side by the hot blank material and cooled down on the outlet by water, oil or air. This causes an inconsistent thermal distribution in the die. The hot extrusion section volume is also cooled down outside of the forming area. In the forming area, the temperature distribution of the hot extrusion sections could be seen as even. This is because the hot extrusion section will be warmed up in an oven prior to the extrusion process.

Contrary to the hot production process of the extrusion section, the die is manufactured at room temperature. The die is in a cold state. The design data and information is also based on the cold state of the die. An enhancement in the optimised die development process is that the first design geometric is not an exact design of the die. It includes a boundary form of the die and an approximated position of the negative hot extrusion section form. This first design should provide the geometrical base to generate a simulation for the thermal distribution of the hot die volume. Hence a steady state simulation of the thermal distribution of the die is achieved. The temperature of the hot blank material is integrated into the simulation as thermal load and the cooling of the die as convection in the environment. The temperature of

the hot blank material and the convection caused by cooling is ascertainable based on prior experiences or measurements on pre-developed machines and projects. The steady state simulation means that the energy inflow is in balance with the energy outflow of the simulation. The results obtained from the thermal distribution in the hot die volume can be used to calculate the deformation caused by the thermal loads. So, a subsequent structure simulation of the first die design includes the thermal distribution over the volume of the die. The result of the structure simulation is the deformation of the first and cold die design caused by the thermal loads of the production process. The deforming of the first die design geometric model (available as a CAD model) results in the geometrical form of the die design, (a derived CAD model) in the hot state. Simulation and CAD modelling operations are accomplished with NX applications.

With the previous simulations, the first die design and the extrusion section in the hot state of the extrusion section production process are achieved. The hot first die design and the hot extrusion section are now available as CAD geometrics. With a Boolean operation, the hot extrusion section geometric can be subtracted from the hot first die design geometric. This Boolean operation produces a new die design. The new hot die design is also available as a CAD model because it is generated with CAD functionalities. The new hot die design includes the hot extrusion section as negative form. This achieves the forming of the hot extrusion section at the production.

The next task is to build a CAD model of the new die design useable for the manufacturing of the die at room temperature. The thermal difference between the hot state (extrusion section production process temperature) and the cold state (die manufacturing temperature) cause a deformation of the die. The new die design in the cold state is achievable by a structure simulation based on the geometry of the new hot die design. The temperature distribution of the die at the extrusion section production process, determined in the first die design simulation, is used as initial temperature. The room temperature of the die manufacturing is the thermal load. The result of this structure simulation is the deformation of the die caused by the temperature difference. In order to reach the cold new die design, the hot new die design has to be geometrically deformed. So, the simulated geometrical deformation results are mapped onto the CAD model geometric of the hot new die design. The CAD functionality deforms the CAD model of the hot new die design based on the simulated deformation. This functionality is used to generate the CAD model geometry of the cold new die design. The resulting data is required for the production of the die.

This improved development process is shown in Figure 5-18. The development process is work done by the author of this these when carrying out consultancy work for the customer (Mahler, Presentation at Wieland Werke, 2010).



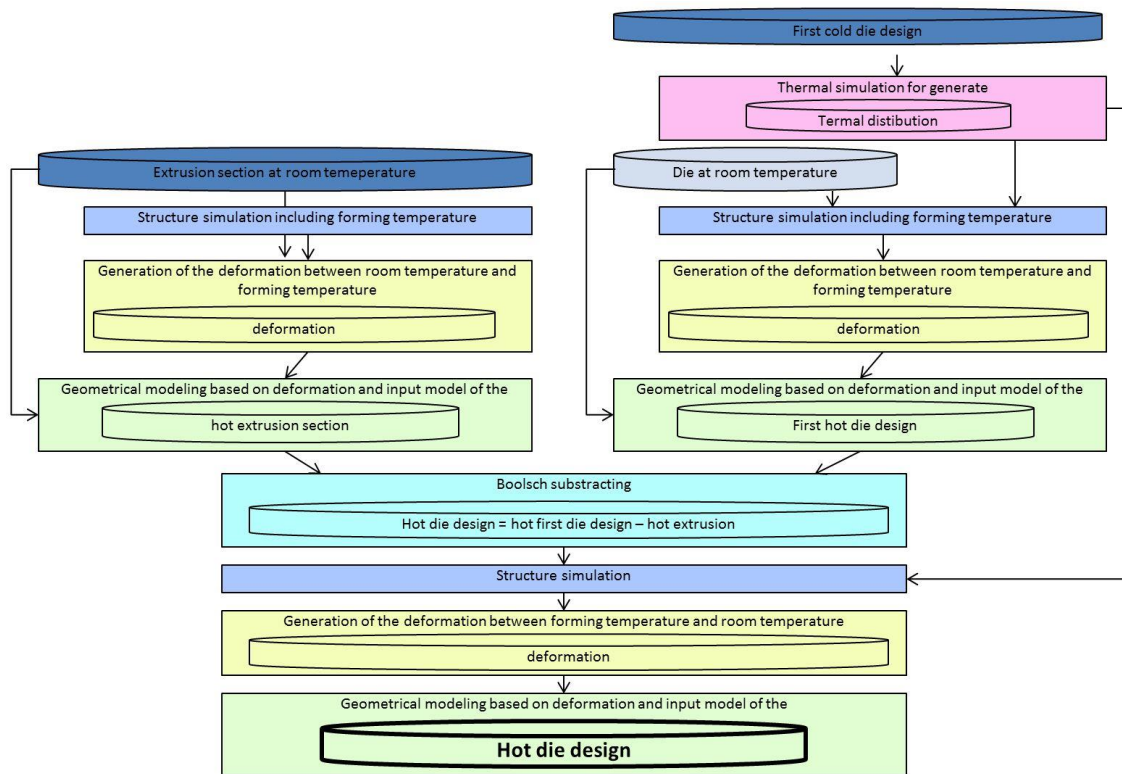


Figure 5-18 Simulation Process of Case 2

### 5.3.3 DISCUSSION ABOUT THE USED AND GENERATED FILES, DATA AND INFORMATION OF CASE 2

In the previous section, it was discussed that two different simulation disciplines are required - thermal and structure simulation. Thermal simulation is used to simulate the temperature distribution in a volume. Structure simulation is used to simulate the deformation of a volume. In some cases, the temperature distribution of the thermal simulation result is used as the input for the structure simulation. This kind of simulation does not require a co-simulation because the thermal and structure simulation run serially. The deformation result is used to deform the geometry stored as CAD model. Similar to the simulation process, the geometrical manipulation of the CAD models runs serially. The CAD manipulation runs after the serial process of the simulation. In conclusion, the new development process of the die is a serial process. Parallel starts are possible from the following two starting bases:

- (1) The CAD model file of the extrusion section and
- (2) The CAD model file of the first die design.

As discussed before, it is possible to start with the CAD model of the extrusion section or with the CAD model of the first die design. Alternatively, it is also possible to start with both of them in parallel way. The following discusses the procedures of both starting points:

- (1) Starting with the CAD model of the extrusion section: The geometry of the extrusion section included in the CAD model file is idealised for meshing. This results in an idealised part (so-called in the used simulation tool NX Advanced Simulation). The idealised part provides the geometry for the meshing. The mesh is stored in a mesh part. The mesh part is used as a base for a simulation part. In this simulation part, the settings for a structure simulation, the simulation parameters, load cases and constraints are added. Based on these settings, the deformation of the extrusion section is calculated. This deformation results from the temperature difference between the cold (expected geometrical form of the extrusion section at room temperature) and the hot state (temperature of the extrusion section at the production). The solve run generates multiple result files which are viewable via the NX post-processor. This post-processor integrates the functionality to generate a deformation-matrix-file of the extrusion section. With the CAD functionalities of NX, the deformation-matrix-file is importable and re-used to determine the changes of the geometry of the cold extrusion section. This new geometry describes the hot extrusion section geometry. The changed geometry is stored as a new CAD model file. This serial process is shown in Figure 5-19.
- (2) Starting with the CAD model of the die design: The geometry of the first die design included in the CAD model file is idealised for meshing. This results in an idealised part. The idealised part provides the geometry for the meshing. The mesh is stored in a mesh part. The mesh part is used as a base for a simulation part. This simulation part includes the settings for a thermal and structure simulation. The thermal simulation has to be run first to generate the thermal distribution result of the first die design during the extrusion section production process. This simulation includes thermal boundaries such as convection in the environment (cooling of the die) and the thermal load caused by the hot material (heating of the die). The solve run generates multiple result files which are viewable via the NX post-processor. This post-processor integrates the functionality to generate a thermal-distribution-file of the die and this result file is re-used as thermal load in the structure simulation. Accordingly, the structure simulation has to be run after the thermal simulation. The structure simulation generates deformation results of the die caused by the thermal difference between the cold (room temperature where the die is manufactured) and the hot state (temperature of the die at the extrusion section production). The solve run generates multiple result files which are viewable via the NX post-processor. This post-processor integrates the functionality to generate a deformation-matrix-file from the cold first die design. With the CAD functionalities of NX, the deformation-matrix-file is importable and re-used to deform the geometry of the cold first die design. This new geometry describes the hot first die design geometry. The deformed geometry is

stored as a new CAD model file. This serial process is shown in Figure 5-19.

These two starting points can be started independently from each other. So, the two starting points can be handled in parallel. However, the process itself is serial because the steps are made serially and the two starting points are achievable because they are independent from each other. The on-going work is also serial and combines the two arms of the starting points. The two new generated CAD models, describing the hot extrusion section and the hot first die design, are assembled into a new third CAD model. The new third CAD model is used for a Boolean operation. This Boolean operation subtracts the hot extrusion section from the hot die design. The result of the Boolean operation is the hot die design geometry with the negative form of the hot extrusion section. This hot die design can form the hot extrusion section in the production process. The geometry of the hot die design is idealised for meshing. The idealised part provides the geometry for the meshing. The mesh is stored in a mesh part. The mesh part is used as the base for a founded simulation part. This simulation part includes the settings for a structure simulation. This structure simulation generates deformation results of the new die design between the hot and cold states. So, the temperature-distribution-file developed during the thermal simulation of the first die design is re-used as the initial thermal boundary. In addition to the initial thermal boundary (starting temperature), the room temperature during die manufacturing is added as thermal load (end temperature). The structure simulation generates the deformation results of the hot new die design caused by the thermal difference between the hot and cold states. The solve run generates multiple result files which are viewable with the NX post-processor. This post-processor integrates the functionality to generate a deformation-matrix-file of the hot new die design. With the CAD functionalities of NX, the deformation-matrix-file is importable and re-used to deform the geometry of the hot new die design. This new geometry describes the cold new die design geometry. The deformed geometry is stored as a new CAD model file. This serial process is shown in Figure 5-19.

These files, data and information of the modelling tools simulation should be managed by TEAMCENTER. TEAMCENTER data management, for this case example, will be discussed in Section 5.3.4.

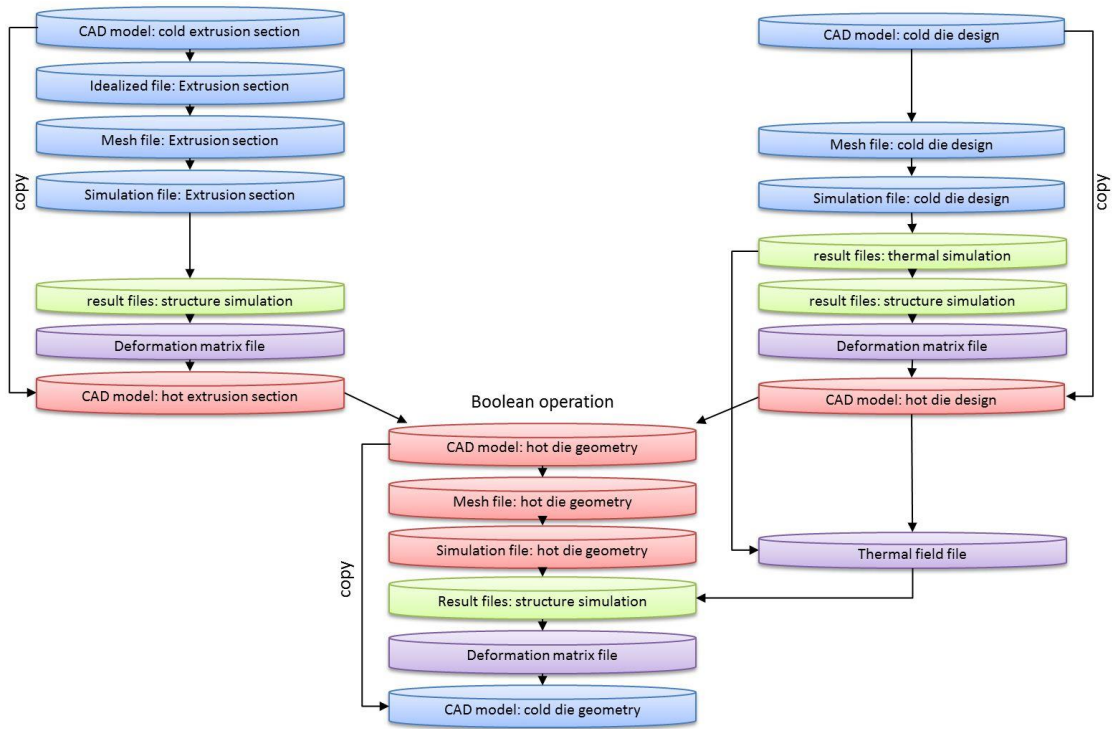


Figure 5-19 File Structure of Case 2

5.3.4 DATA MANAGEMENT OF CASE 2 WITH TEAMCENTER

This case study includes an integration of the system-simulation into the PLM framework of TEAMCENTER. The management of all the required files and the support of the simulation process should be achieved.

The case study starts with two independent CAD models of the cold extrusion section and the first die design. This involves two independent starting points that could run in parallel until the data is combined. The two parallel process arms are similar to each other. Each of the CAD models is managed as a dataset under a CAD item revision. Based on each of the CAD model part files, the idealised part files are generated and managed as datasets under new CAEGeometry item revisions. Each CAEGeometry item revision has a CAESource and CAETarget relationship pointing to the source CAD item revision (Siemens Product Lifecycle Management Software Inc, 2012). The two idealised part files are the source for the two associated mesh files. Based on each of the idealised part files, the mesh files are generated and managed as datasets under new CAEModel item revisions. In order to link the new CAEModel item revision to the appending CAEGeometry item revision and appending CAD item revision, each CAEModel item revision has a CAESource relationship pointing to the appending CAEGeometry item revision and a CAETarget relationship pointing to the appending CAD item revision (Siemens Product Lifecycle Management Software Inc, 2012).

Based on each of the mesh files, simulation files are generated and managed as datasets under new CAEAnalysis item revisions. In order to represent the dependency to the associated mesh file, each CAEAnalysis item revision has a CAEDefining relationship pointing to the appending CAEModel item revision. The simulation-file of the extrusion section includes a structure simulation solution. The simulation file of the first die design includes both a thermal and a structure simulation solution. The result files generated by the solve run are stored in datasets under the appending CAEAnalysis item revision. In the case of the extrusion section, the result files of one thermal simulation are managed. In the case of the first die design, the result files of one thermal and one structure simulation are managed. The post-process of the cold extrusion section results produces an additional deformation file. This deformation file is managed as a dataset under the CAEAnalysis item revision of the cold extrusion section and is re-used later for CAD model deformation functionality. The post-process of the cold first die design generates an additional thermal distribution file. The temperature distribution file is managed as a dataset under the CAEAnalysis item revision of the first die design simulation. This file is re-used during later simulation steps. The post-process of the first die design also generates an additional deformation file. This deformation file is managed as a dataset under the CAEAnalysis item revision of the cold first die design. This file is re-used later for CAD model deformation functionality. In order to deform the CAD model geometry of the extrusion section and the die design, the CAD model files are cloned. The CAD models are available in the cold state. The two cloned CAD model files are managed as datasets under two new CAD item revisions. With CAD functionalities, the two cloned CAD model geometrics are transformed based on deformation files. This transformation re-uses the two deformation files to transform the cold state geometry into a hot state geometry. The cold first die design geometry is transformed (based on the deformation file of the die design) into the hot first die design geometry and the cold extrusion section geometry (using the extrusion deformation file) to the hot extrusion section geometry. The hot state describes the extrusion section production state, the cold state the room temperature state.

However, there are no relationships between the two new generated CAD item revisions and the appending deformation file datasets under the CAEAnalysis item revision. So, this process cannot be uniquely retraced. So sometimes it is required to use a trace-link.

“A trace link establishes a path in which one object takes precedence over another. The trace link creates a directional relationship between the two objects, a relationship conveyed by the terms defining and complying.” (Siemens Product Lifecycle Management Software Inc., 2012).

However, this link is not precise because it is not pointing to the dataset of the CAEAnalysis item revision. Additional information to identify the precise dataset is required. So, an out-of-the-box solution is not available. Customization can be used, but is not representative for general applications. An automated data appropriation is not possible. So, the deformation file datasets of the CAEAnalysis item revisions has to be applied manually.

There is also missing functionality in the case of the first die design. The simulation file includes two sub cases: the thermal and the structure sub case. The op2 result file of the

thermal sub case is used for input in the structure sub case. This is not monitored in TEAMCENTER. This input is achievable with NX and TEAMCENTER because the necessary data is available. Unfortunately, this process is not retraceable in the data management level.

The described data management is shown in Figure 5-20.

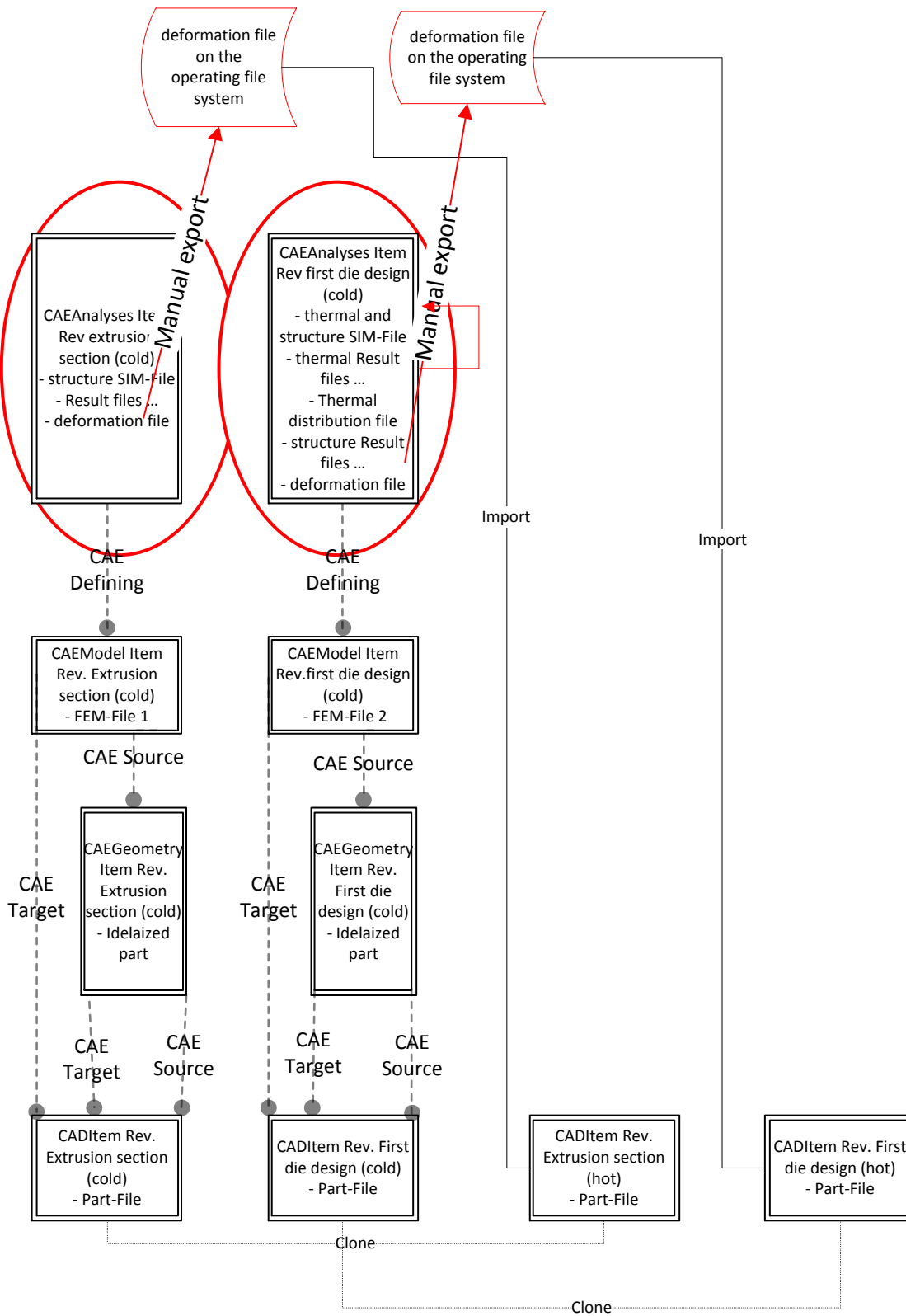


Figure 5-20 Metadata Structure of Case 2 Phase 1



The two new CAD models are assembled into a new third CAD model. This CAD model includes the geometry of the hot extrusion section and hot die design managed as a dataset under a new CAD item revision. With a Boolean operation, the hot extrusion section is subtracted from the hot die design. The result of the Boolean operation is a hot die design geometry with the negative form of the hot extrusion section. This hot die design can form the hot extrusion section in the production process. The new CAD model of the hot new die design is the source for meshing. The generated mesh files are managed as datasets under a new CAEModel item revision. In order to link the new CAEModel item revision to the appending CAD item revision, the CAEModel item revision has a CAESource relationship and a CAETarget relationship pointing to the appending CAD item revision (Siemens Product Lifecycle Management Software Inc, 2012). Based on the mesh file, a simulation file is generated and managed as a dataset under a new CAEAnalysis item revision. In order to represent the dependency to the depending mesh file, the CAEAnalysis item revision has a CAEDefining relationship pointing to the appending CAEModel item revision. The simulation file includes a structure simulation solution. As explained in Section 5.3.3, this structure simulation requires the thermal distribution file of the cold die design as input. A CAEInclude relationship points from the new CAEAnalysis item revision to the CAEAnalysis item revision of the cold first die design where the thermal distribution file is included as a dataset. The CAEInclude relationship can help to appropriate the thermal distribution file manually or automatically. However, the CAEInclude relationship points to the CAEAnalysis item revision, not to the dataset. In this case, it is easier to make the appropriation manually. The solve run generates multiple result files managed as datasets under the CAEAnalysis item revision. The post-process of the hot new die design results produces a deformation file. This deformation file is managed as a dataset under the CAEAnalysis item revision of the hot new die design. This file is re-used later for CAD model deformation functionality. The CAD model of the hot new die design is cloned to a new CAD model managed as a dataset under a new CAD item revision. This CAD model is used to generate the geometry of the new die design in the cold state. With CAD functionalities, the geometry of the new CAD model is geometrically transformed based on the deformation file of the structural hot new die design simulation. This transformation re-uses the deformation files to transform the hot state geometry into a cold state geometry. The hot new die design geometry is transformed (based on the deformation file of the hot new die design) into the cold new die design geometry. The hot state describes the extrusion section production state and the cold state, i.e., the room temperature state where the die is manufactured. However, there is no relationship between the two new generated CAD item revisions and the appending deformation file datasets under the CAEAnalysis item revision. So, this process is not uniquely retraceable. There is the possibility to use a trace-link. However, this link is not precise because it is not pointing to the dataset of the CAEAnalysis item revision. Additional information is required to identify the precise dataset. So, an out-of-the-box solution is not available. Customization can be used, but is not suitable for general use. An automated data appropriation is not possible. So, the

deformation file datasets of the CAEAnalysis item revisions have to be appropriated manually.

The described data deformation management is also shown in Figure 5-21.

In summary, TEAMCENTER provides good support and management of individual simulation models. In so doing, traceability exists in an individual simulation model from its sources to its significant product parts. Nevertheless, TEAMCENTER does not support multi-disciplinary and inter-disciplinary interaction of individual simulation models. The unique traceability of multi-disciplinary and inter-disciplinary interaction will not be achieved and required data will neither be identifiable or available. Consequently, a unique review of multi-disciplinary and inter-disciplinary interactions between individual simulation models with TEAMCENTER is not possible.

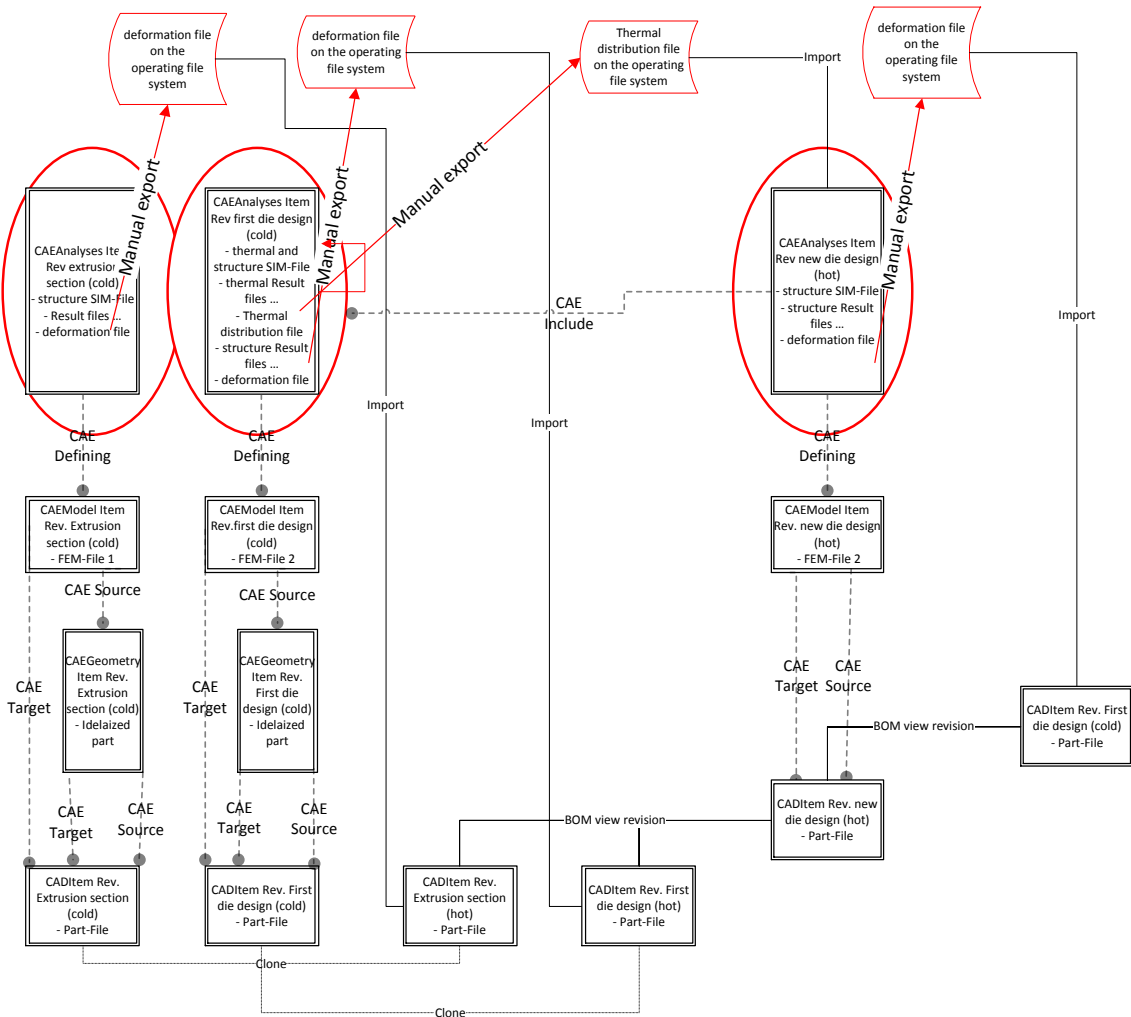


Figure 5-21 Metadata Structure of Case 2

## 5.4 CASE 3: SYSTEM-SIMULATION OF A LIFT GATE

An automotive company in the south of Germany sought a benchmark to identify the best PLM and CAx system (Computer Added system with x standing for the multi functionalities like design [CAD], analysis [CAE] and manufacturing [CAM]) for their companies. The benchmark was partitioned into different projects and work packages. One project was called 'future engineering'. In the 'future engineering' project, the work package 'system-simulation of a car luggage door' was involved. The simulation of the lift gate was to represent and include the mechatronic systems of the lift gate product. The result of the simulation should be to generate knowledge about the functionality and parameter values describing the lift gate product. So, this system-simulation required a linking of different simulation models and solvers coupled to a co-simulation. The lift gate is shown in Figure 5-22.

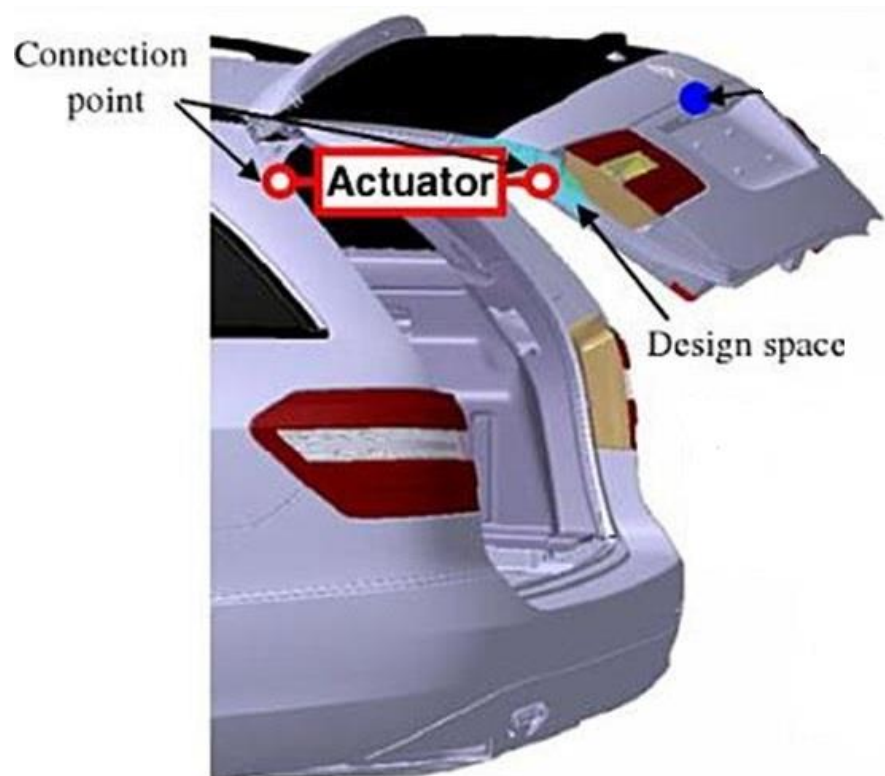


Figure 5-22 Car Lift Gate (Wuttke, Bohn, & Suyam-Welakwe, 2011)

### 5.4.1 DISCUSSION ABOUT THE LIFT GATE SYSTEM

Although, the lift gate is only a subsystem of the car, in this example it is considered a system in its own right. The lift gate itself is built on multiple subsystems. The system, the

subsystems and the dependencies between the subsystems are shown in Figure 5-23 and discussed in this section.

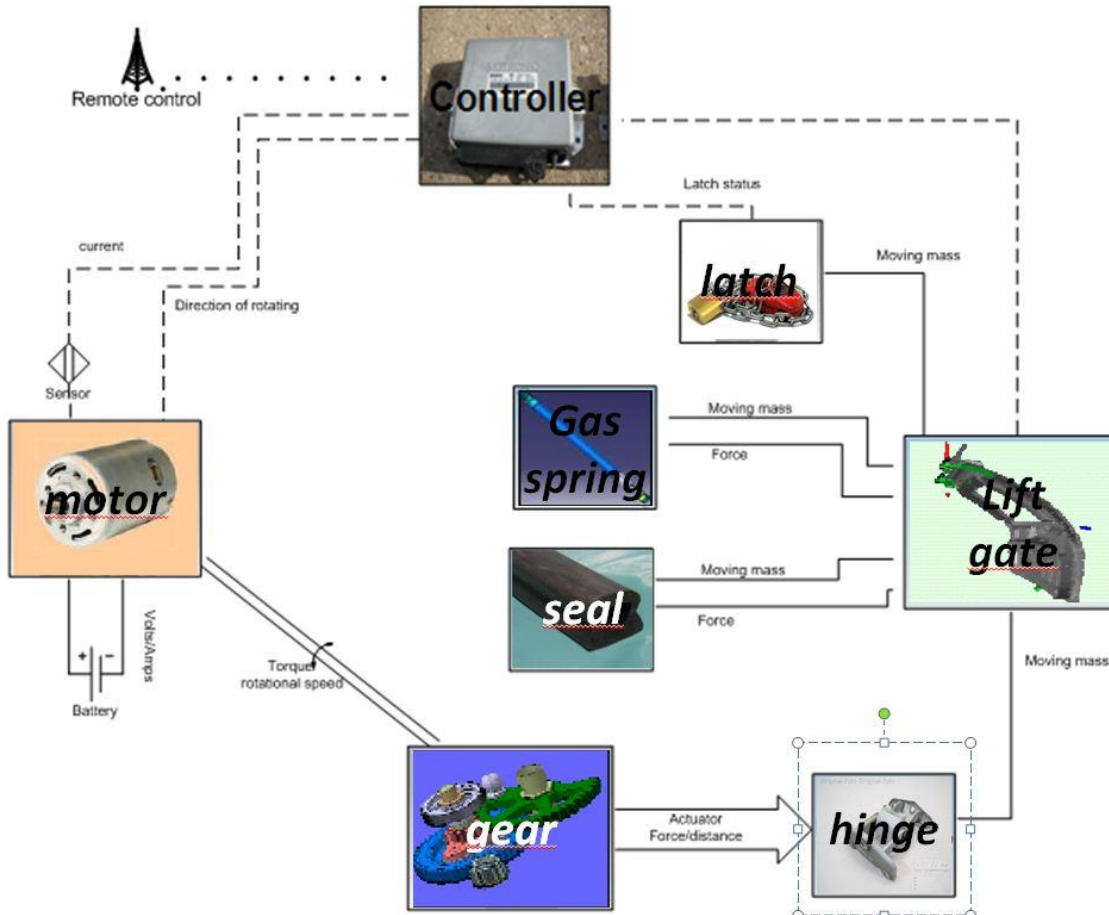


Figure 5-23 Lift Gate System Architecture

Each subsystem of the lift gate can be run on its own. So, the simulation of subsystems should be run on its own. Due to this requirement, the missing inputs of the bordering subsystems have to be predefined as the input values. In the following, the individual subsystems and the interaction of the subsystem in the lift gate system are discussed:

1) Controller subsystem:

The controller controls the electrical motor and the latch. The controller receives information about the latch status. So, the controller 'knows' if the latch is in the opened or closed state. The closed state of the latch prohibits the powering of the electrical motor. The controller also receives incremental angle information from the incremental sensor of the electrical motor and the used current of the electrical motor. This information is used to detect impacts during the movement of the lift gate. So, the change of the current and the angle change are interpreted by the software of the controller. Based on this information, the controller controls the electrical motor.

- 2) Electrical motor (Sui & Hirshey II, 2000;Boberg, 2008):  
The electrical motor offers drilling speed and momentum. The connection to the electrical power is guided and controlled by the controller. The electrical motor transforms electrical energy to rotational energy that is initiated in the gear system.
- 3) Gear subsystem (Boberg, 2008):  
The gear transforms the rotational energy of the electrical motor into a translator energy and generates a three dimensional movement of the connecting rod. This energy transformation provides a low speed, high force energy as output by transforming the high speed and low momentum energy input of the electrical motor. So, the gear system transforms the rotation speed and momentum energy and provides output to the connecting rod converting the rotational energy into a transferable energy. This connecting rod is ultimately linked to the hinge system.
- 4) Hinge subsystem:  
Two hinges are available to connect the door to the chassis. Each of the hinges is divided into two main parts; one of them is fixed to the chassis and the other to the door. These two parts are linked together by a joint that provides a rotational movement around the joint axis. The hinges enable the lift gate door to rotate around the joint axis. One of the two hinges, the part that is fixed to the lift gate door, is also linked to the connecting rod of the gear system. This connecting rod initiates the transferable energy (which includes force and movement of the connecting rod) to this part of the hinge. In this way, the transferable energy of the connecting rod can force a movement of the lift gate door.
- 5) Lift gate door subsystem (Boberg, 2008):  
The lift gate includes all the bodies and parts that have to be moved by the system to open the trunk. By using the bodies and parts, the required mass and inertia of the lift gate can be abstracted. The lift gate door is fixed to the hinge system. Due to the forced movement of the second part of one hinge, the lift gate is forced to open or close. Moreover, a gas spring is fixed to the lift gate. Additionally, the lift gate door can be in contact with the seals. The lift gate door can be fixed by the latch system.
- 6) Seal subsystem:  
The seal subsystem binds water ingress to a lagged area. It also dampens noise and acceleration resulting in relative movements between the chassis and the lift gate door. The damping effect of the seals between the lift gate door and the chassis cause a force when the lift gate is at the nearly closed or the closed position. The seals also act like a spring. The spring force orientation of the seals lifts the lift gate door. The seals are fixed to the chassis but can also be in contact with the lift gate door.
- 7) Gas spring subsystem (Sui & Hirshey II, 2000):  
On one side, the gas spring system is connected to the chassis and on the other side to the lift gate. The gas spring reacts with forces dependent on the relative position and relative velocity between the lift gate door and the chassis. The force, dependent on the velocity, causes damping effects to the moving lift gate door and, dependent upon the position, causes spring effects.

## 8) Latch subsystem:

The latch influences, the possible movement of the lift gate. In the closed state it fixes the lift gate to the chassis. It fixes its position in the closed state to the chassis by a hook that is in contact to an eye fixed to the chassis. In this state, the opening of the lift gate is impossible. In contrast, the open state is a disconnected state between lift gate door and chassis, which allows movement of the lift gate door. In the closing or opening procedure, the contact between the hook (connected to the lift gate door) and the eye (connected to the chassis) changes their contact force. The force appears at the nearly closed or closed position of the lift gate. The latch forces orientation closes the lift gate. The latch forces work against the seal forces.

Each subsystem is designed to work its own. A huge number of the subsystems will constitute supplier parts. This is illustrated in Figure 5-24. The integration of a supplier into the system-simulation of the lift gate is not part of the benchmark required by the automotive company but is an important aspect in the planning of future multi-disciplinary data management tools. Nybacka, Törlind, Larsson & Johanson (2006) discussed a concern of companies to implement 'black-box-simulations' of suppliers into system simulations. The suppliers should generate simulation models of the subsystem they deliver. The guarantor of the entire system uses subsystem-simulation models and integrates them into their system-simulation model. Usually, suppliers would keep their simulation secret. The simulation model includes know-how that shouldn't be publicised to other companies or persons. So, the simulation model should be a 'black-box-simulation' excluding know-how of the companies. In Link (2012), the integration of possible 'black-box-simulations' using the MODELISAR FMI (functional mockup interface) interface is shown in Figure 5-24. The future of lift gate simulation integrates simulation models of suppliers, but not the actual system-simulation example.

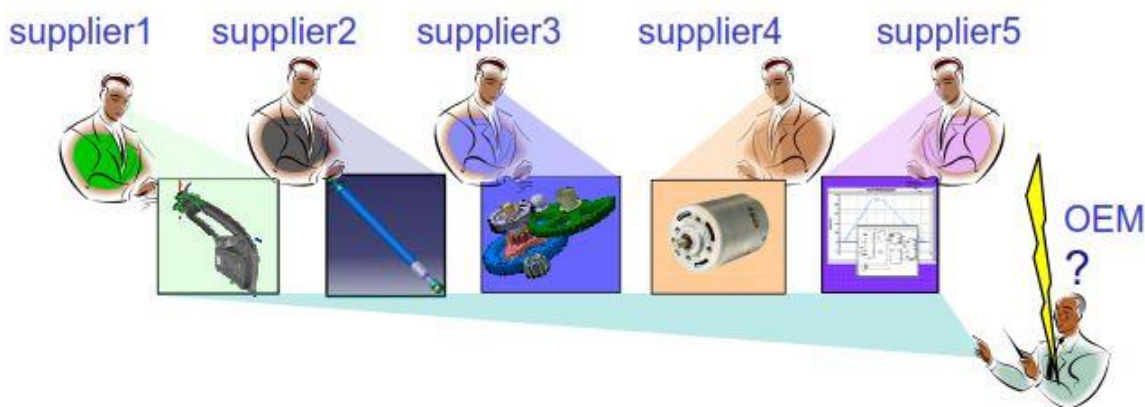


Figure 5-24 Supplier Integration (Link, 2012)

In Section 5.4.2, the simulation process and information flow of the lift gate system will be discussed.



#### 5.4.2 DISCUSSION OF THE SYSTEM AND SUBSYSTEM-SIMULATION OF CASE 3

In order to keep the subsystems separate and to support a subsystem-oriented development process, each subsystem should be generated as an independent simulation model. So, each subsystem has its own simulation model. In order to build the lift gate system-simulation, the subsystem-simulation models are merged or connected. The merging and connecting is completed in two steps:

- (1) The first step is to merge subsystem-simulation models that are generated in the same simulation authoring tool into a merged simulation model.
- (2) The second step is to connect the merged simulation model and the unmerged subsystem-simulation models to a system-simulation by coupling the different simulation models and solvers. The coupling of the solvers is necessary to execute simulation models of different simulation authoring tools.

For the generation of the subsystem-simulation models, a useful simulation discipline has to be identified for each subsystem. The simulation discipline required for this system-simulation can be specified in flexible bodies, rigid bodies and logical simulations (Mahler, 2012):

- (1) Flexible bodies are bodies where the self-deformation of the body is important (such as the deformation of a softball dropping to the ground).
- (2) Rigid bodies will not self-deform. So, the self-deformation has to be negligible (such as the negligible deformation of the ground where a softball has dropped).
- (3) The logical and mathematical oriented elements such as controller or software belong to a logical simulation discipline (Tian, Yan, Parkin, & Jackson, 2008); they will be based on simulated equations.

Based on this hierarchy, subsystems can be ordered to simulation disciplines. The controller subsystem-simulation and the software subsystem-simulation belong to the logical simulation discipline. The electrical motor also belongs to the logical simulation discipline. It has to be controlled based on used current. This control requires a logical controller model. The energy transformation is described by a mathematical model of the electrical motor. Such mathematical descriptions are ideal for equation-based simulation models. So, EBS models can be structured to a logical simulation discipline. The self-deformation of the gear system, latch and lift gate system is negligible. So, these subsystem-simulation models can be generated with a rigid body simulation discipline. Such rigid body simulation models are usually generated with MBS tools. In contrast, the self-deformation of the seals and gas spring system is significant. In these cases, the resulting effects like forces of the self-deformation are known. These effects are summarised in fields that describe spring forces or damping forces dependent upon self-deformation or moving velocity. The resultant knowledge of the seal-subsystem-spring-curve, gas-spring-subsystem-spring-curve and gas-spring-subsystem-damping-curve can be used in the simulation. The curves can be used in rigid body simulation disciplines as nonlinear spring or damping curves. Based on these



curves, the deformation or movement of the gas-spring-subsystem or seal-subsystem is calculated. In contrast to seal and gas-spring subsystems, the self-deformation of the hinge system is important. The actual deformation of the hinge is small but has a big influence on the movement of the lift gate. Even though the hinge self-deformation is minimal, the lift gate oscillates. Due to the lift gate arm of the lever, the self-deformation of the hinge is transformed to a higher movement at the lift gate end. In order to integrate this self-deformation, the hinge has to be modelled as a flexible body (Mahler, Vickeres, Hasse, Traulich, Schmerr & Hitzer, 2010). Flexible body simulation disciplines are ideally generated using finite element analysis (FEA). The role of the subsystems within the simulation discipline is shown in Figure 5-25.

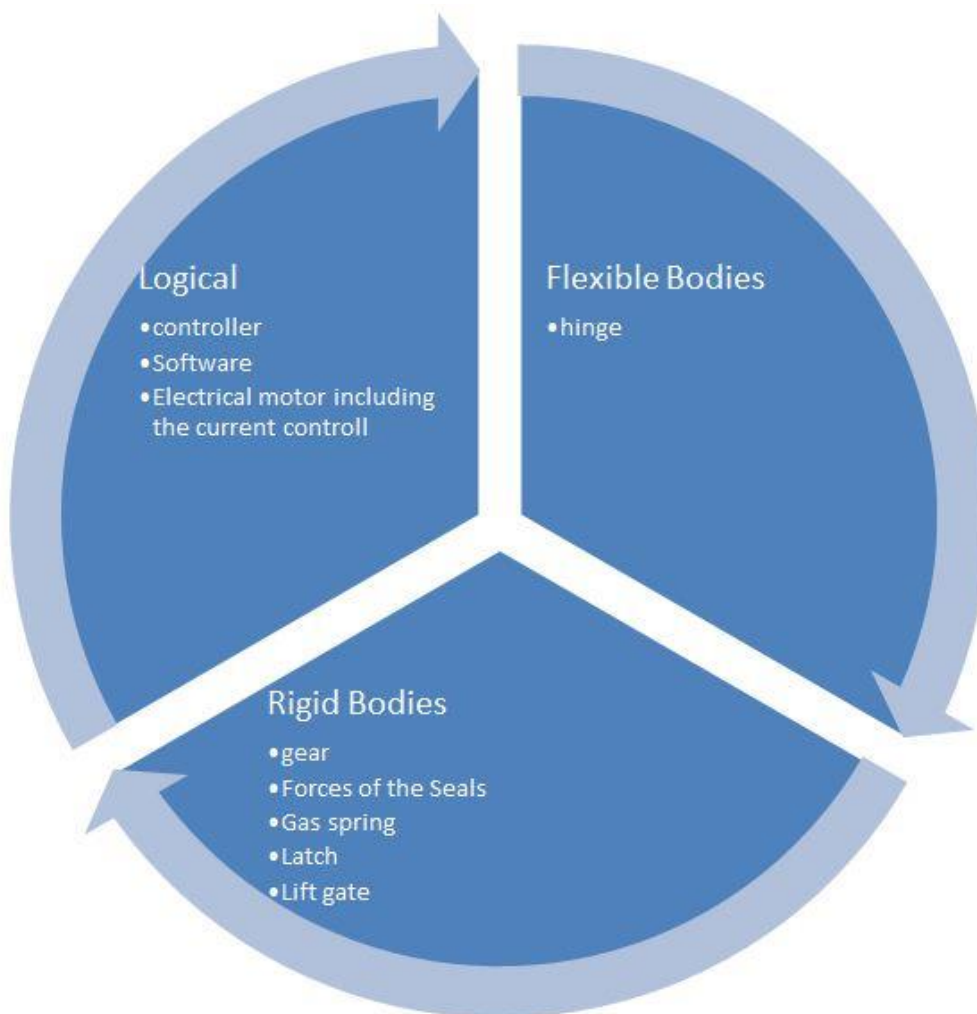


Figure 5-25 Simulation Discipline Architecture Case 3

The three simulation disciplines (flexible body, rigid body and logical simulation) can be achieved with the tools described in Table 5-2 and Figure 5-26.

Table 5-2 Case 3: Table of Simulation Tools

Simulation Discipline	Simulation Tool
Flexible body simulation	NX Advanced Simulation and NX Flexible Body
Rigid body simulation	NX Motion and add on NX Motion Control
Logical simulation	Matlab and add on Simulink

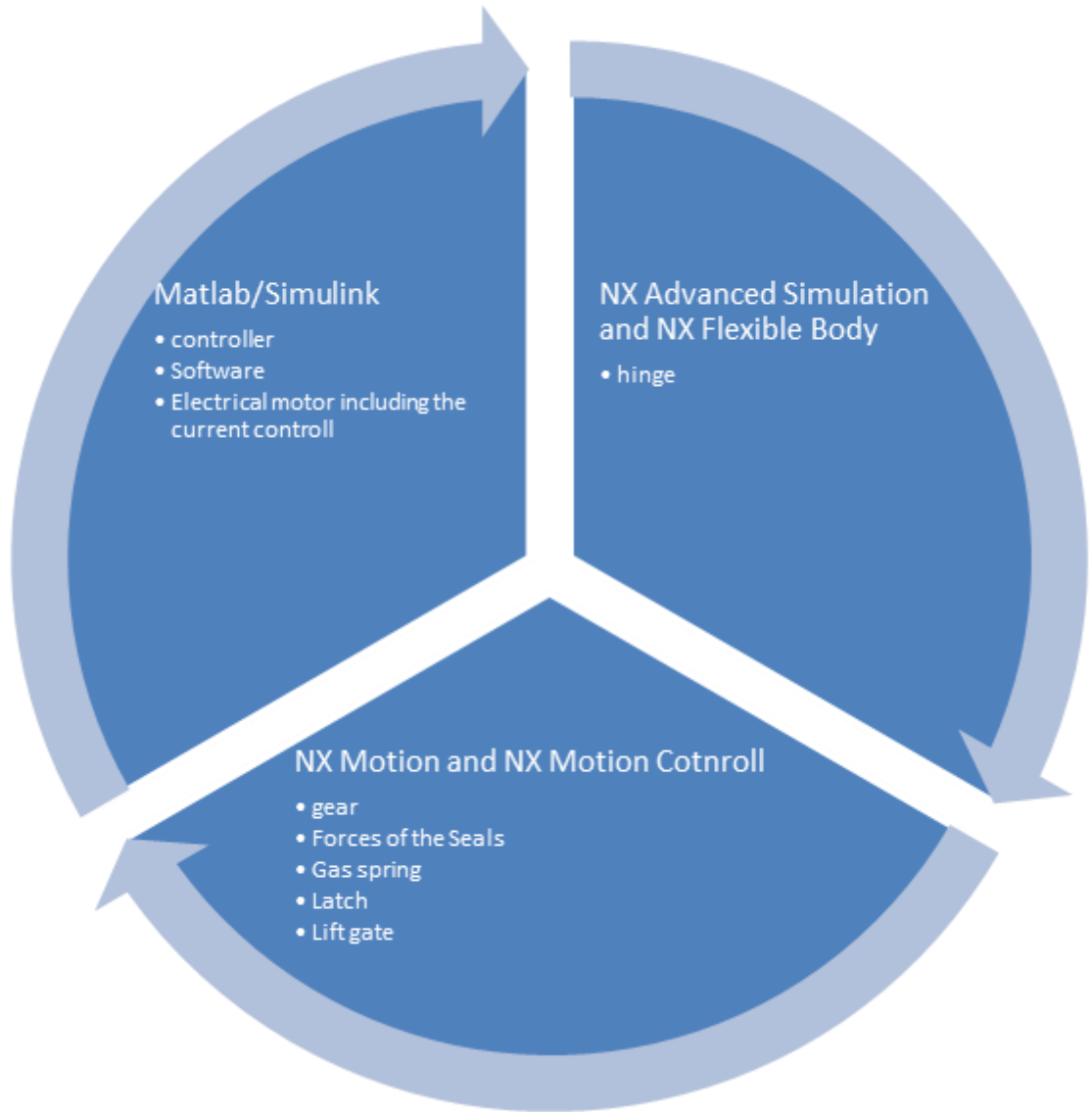


Figure 5-26 Case 3: Simulation Tool Architecture Source

The discussion regarding simulation generation of each subsystem is protracted and the details are irrelevant for the case study. So, Section 5.4.3 will focus on a discussion regarding simulation architecture and system-simulation generation.

### 5.4.3 DISCUSSION OF THE SIMULATION ARCHITECTURE AND SYSTEM-SIMULATION GENERATION OF CASE 3

This section discusses the simulation architecture and the generation of the system-simulation. The first three parts will elaborate on the data architecture of each simulation discipline used in this case example. Afterwards, Section 5.4.3 IV explains the assembling of subsystem-simulation models to the system-simulation.

#### I. Architecture of the Logical Simulation Discipline of Case 3

In order to generate the logical simulation models of the case example, Matlab/Simulink is used as the simulation authoring tool. The subsystem controller, software and electrical motor are modelled and solved using the simulation tool Matlab/Simulink. This simulation model was prepared by the aforementioned automotive company. The three subsystems were not modelled as separate simulation models. A summarised simulation model of these three subsystems was provided by the same automotive company. This simulation model is stored in a Matlab/Simulink mdl-file-format.

So, the logical simulation discipline does not include the architecture of simulation models or data because it was provided externally in a prepared, single format.

#### II. Architecture of the Rigid Body Simulation Discipline of Case 3

The simulation models of the rigid body simulation were implemented with the MBS tool, NX Motion, from Siemens Industry Software. Simulation models of the gear, gas spring, latch, lift gate and seal subsystem were generated. Each of these subsystems was modelled as an independent simulation model. The pre-modelled CAD geometry parts were provided by the automotive company as input for the independent simulation models. In Table 5-3 the independent simulation models are listed with predefined input-data, output and modelling annotations.

Table 5-3 Independent Rigid Body Simulation Models of Case 3

Subsystem	Input	Modelling Annotation	Output
Gear	CAD geometry	NX Motion can read all necessary information to define the rigid bodies from the CAD geometry. The joints to link the rigid bodies can be defined based on the CAD geometry. The different gear ratios between gear wheels were determined from the CAD geometry.	Gear system-simulation model
Gas spring	CAD geometry, gas spring fact sheets	NX Motion can read all necessary information to define the rigid bodies from the CAD geometry. The joints to link the rigid bodies can be defined based on the CAD geometry. The spring parameters as well as the damping parameters were determined from the gas spring fact sheets.	Gas spring system-simulation model
Latch	CAD geometry	NX Motion can read all necessary information to define the rigid bodies from the CAD geometry. The joints to link the rigid bodies can be defined based on the CAD geometry. The latch includes an electrical motor for energy generation. This requires a controller. The latch should not be detailed in depth. So, the control of the motor was modelled by a signal and PMDC-motor without a factual conclusion. The control of the signal and the electrical description of the PMDC motor are modelled. The control signal to start the engine manually is stored in a separate AFU-file which is readable by NX. This signal substitutes the control of the latch.	Latch system-simulation model, AFU-file to substitute the control of the latch
Lift gate	CAD geometry	NX Motion can read all necessary information to define the rigid bodies from the CAD geometry. The joints to link the rigid bodies can be defined based on the CAD geometry.	Lift gate system-simulation model
Seal force	CAD geometry, measured spring force curve by a seal test	NX Motion is able to add additional design objects to the existing CAD geometry base. This is used to model two lines. The seals chassis side as well as the lift gate door side is represented by these two lines. Between these two lines, multiple springs are modelled. The springs have a non-linear spring reaction force dependent on the deformation. The spring force curve defines the force as null in the case of a seal positioned without contact between seal and lift gate door. In the position where the seal is in contact with the lift gate door, the seal force curve is generated based on a measured spring force curve. This resulting spring force curve is stored in a separate AFU-file which is readable by NX.	Seal force system-simulation model, AFU-file describing the spring-force-curve

### III. Architecture of the Flexible Body Simulation Discipline of Case 3

The hinge has to be simulated as a flexible body. The self-deformation of the hinge is transformed by the lift gate arm of the lever to a much higher movement. This movement has to be analysed by integrating it into the simulation. So, the self-deformation of the hinge has to be considered which requires the modelling of the simulation as flexible body. The generation of the simulation model is made using the FEA tool NX Advanced Simulation from Siemens Industry Software. The hinge consists of two main parts linked together by a bolt. In the case of the hinge, a decision has to be made as to whether both main parts of the hinge are required as flexible bodies or only one of them. The decision was to integrate only the main part connected to the lift gate door as a flexible body. This was not a technically verified decision but one where it was deemed necessary to keep the work minimal so as to present the capabilities of the software. The flexible body requires a rigid body simulation model to be capable of being integrated into a co-simulation, i.e. a simulation model for the flexible body has to be generated which makes it possible to be coupled to a predefined rigid body simulation model. So, the meshing and generation of the flexible body have to respect predefined points. These points are also predefined for the MBS simulation model. In the FEA simulation model, these points are transformed to nodes and linked or integrated with the mesh of the hinge part. Moreover, free or fixed boundary degrees of freedom are added as constraints to these nodes. The nodes and the constraints of the flexible body simulation model, and the points of the rigid body simulation model, provide the opportunity to act as communication points (Siemens Product Lifecycle Management Software Inc., 2011).

After generating the FEA simulation model, the pre-processor is able to generate an input deck for the FEA solver. Using this input deck, it is possible to execute the solve run.

The solve run reduces the mathematical matrix to a fast operational dimension that maintains the representation of the simulation model. The result is a flexible body reduced simulation model matrix of the hinge (Siemens Product Lifecycle Management Software Inc.) in the rfi-file-format. In NX Motion, a rigid body can be replaced by a flexible body. So, the communication points have to be congruent and the flexible body has to be expressed as a reduced simulation model matrix such as in the rfi-file-format (Nowakowski, Fehr & Eberhard, 2011). This replacement of the rigid body by a flexible body is solvable by the interaction between the MBS and FEA solver of NX.

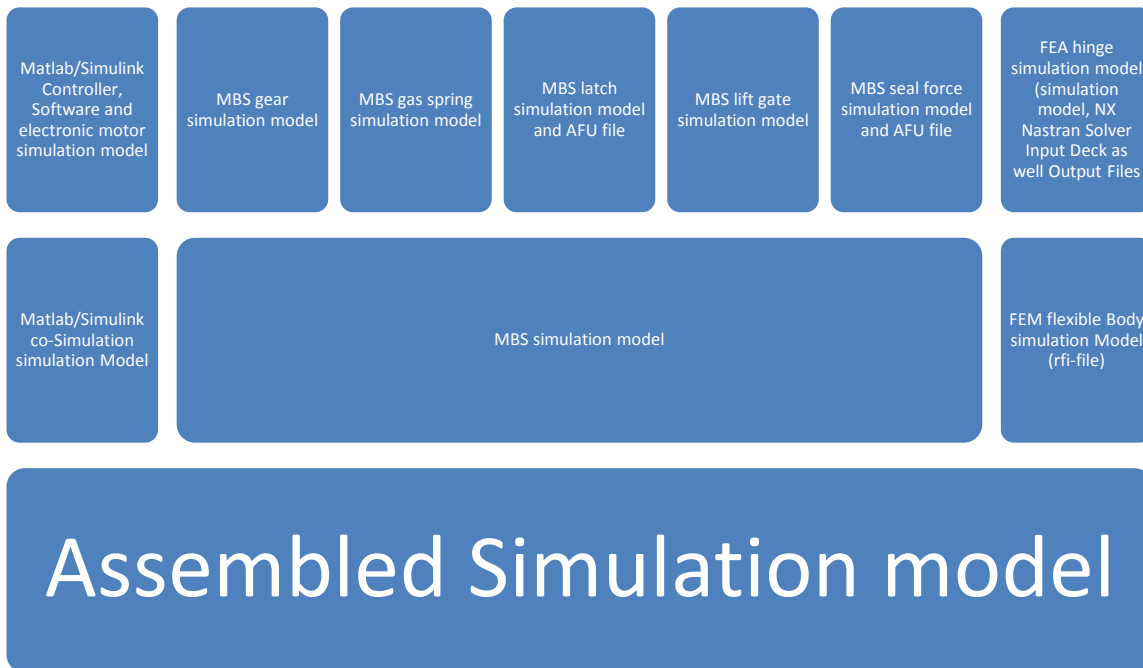
### IV. Assembling of the Simulation Models of Case 3

The assembling of different simulation models and simulation disciplines is carried out in multiple steps. These steps are dependent on the simulation discipline. In some simulation disciplines, the merging of simulation models is possible. This will be discussed for each simulation discipline:

- Flexible body simulation discipline:  
The hinge is considered to be in the flexible body simulation discipline. The solve run of the finite element simulation model generates the necessary rfi-file. In order to get a corresponding flexible body, predefined points are integrated into the flexible body as communication points to a rigid body simulation model. This rfi-file is used to exchange a rigid body by a flexible body represented by the rfi-file. In the current example, the rigid body representing the second part of the hinge is replaced by the flexible body described by the rfi-file.
- Rigid body simulation discipline:  
The five individual MBS simulation models, representing the gear subsystem, lift gate body subsystem, gas spring subsystem, latch subsystem and seal subsystem, can be merged into one MBS simulation model. The individual MBS simulation models are appended to the product structure. The sub-ordered MBS simulation models in the product structure can be merged into higher-ordered MBS simulation models in the product structure. These sub-product structures, representing the subsystems, are included at the top product structure of the lift gate system. An MBS simulation is created at the top of the product structure with NX Motion. Based on the product structure and subordinate subsystems, NX Motion can identify the dependent MBS simulation models of the subsystems. The MBS simulation models of the subsystems can be integrated to the top MBS simulation using the ordering of the product structure. The links, joints and parameters of the sub-simulation models are merged into the top simulation model (Siemens Product Lifecycle Management Software Inc., 2011). A few of special simulation information cases of the top MBS simulation, require additional work to be done to rebuild missing or lost imported simulation information.
- Logical simulation discipline:  
The automotive company provides a summarised Matlab/Simulink simulation model including all the subsystems that should be simulated in the logical simulation discipline. NX Motion Control provides the possibility to link the NX Motion MBS simulation model to a Matlab/Simulink simulation model. So, the Matlab/Simulink simulation model has to integrate an s-function to the Matlab/Simulink simulation model that couples the two solvers to each other and organises the parameter exchange during the co-simulation. The s-function generation is automated and supported by NX Motion Control and Matlab/Simulink. This s-function has to be integrated into the original Matlab/Simulink file which results in a new Matlab/Simulink simulation model. The new generated Matlab/Simulink simulation model is saved as a new mdl-file with predefined name regulations by NX Motion Control. The NX Motion simulation model and the new Matlab/Simulink simulation model are runnable as a co-simulation between NX Motion and Matlab/Simulink.

The assembled simulation model consists of three simulation models: the NX Motion model for the rigid body simulation, the RFI file for the flexible body simulation and the

Matlab/Simulink model for the logical simulation. All three simulation models can be coupled with NX. The coupling makes co-simulation possible between the three solvers of NX Motion, NX Nastran and Matlab/Simulink. The host of the co-simulation can be NX Motion or Matlab/Simulink. The host controls the simulation sampling rate and communication between the three solvers. This system-simulation architecture is shown in Figure 5-27.



**Figure 5-27 Simulation Model Architecture of Case Example 3**

The files, data and information of the lift gate system-simulation should be managed by TEAMCENTER. In Section 5.4.4, the TEAMCENTER data management for this case will be discussed.

#### 5.4.4 DATA MANAGEMENT OF CASE 3 WITH TEAMCENTER

The case study includes the integration of the system-simulations into the PLM framework TEAMCENTER. The management of all the required files and the support of the simulation process should be achieved.

The starting point is the CAD model, hierarchically ordered in CATIA-format, and the provided Matlab/Simulink simulation model. By using NX as a simulation authoring tool, the CAD model has to be transformed and imported into NX-format (Siemens Product Lifecycle

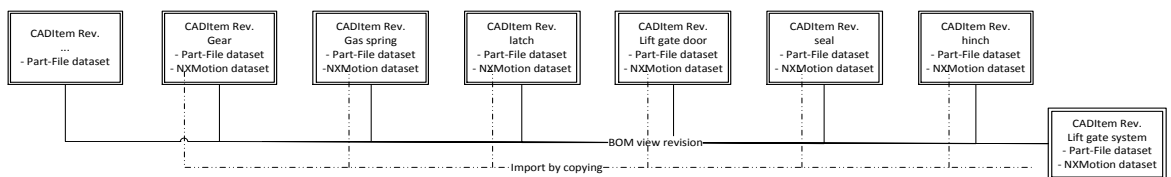


Management Software Inc., 2011). This import also includes the hierarchical ordering of the imported CAD models in NX-format. The CAD models in NX-format are the new base. In the following, only the main parts and structure are included. The CAD models are managed as datasets under CAD item revisions. Similarly, each of the assembly files is stored as a dataset under a CAD item revision and, additionally, a BOM view to suborder the hierarchical CAD model structure (Siemens Product Lifecycle Management Software Inc., 2012).

### I. Rigid Body Simulation Models

The simulation process generates five NX Motion MBS simulation models as shown in Figure 5-27. These five NX Motion MBS simulation models represent: the gear subsystem, the gas spring subsystem, the seal subsystem, the latch subsystem and the lift gate door subsystem. An additional NX Motion MBS simulation model for the hinge subsystem will be created to optimise the integration of the flexible body into the summarised system simulation. The sources for these six NX Motion MBS simulation models are the CAD assembly models of these subsystems. The six CAD assembly models are subordinated under the lift gate system CAD assembly model. The lift gate system CAD assembly model is used for the summarised NX Motion MBS model where the six NX Motion simulation models are merged by automated integration functionalities to a higher-ranked simulation model (described in Section 5.4.4 IV).

The NX Motion simulation model is managed as an NX Motion dataset directly to the appending CAD item revision (Siemens Product Lifecycle Management Software Inc., 2011). The CAD item revision refers to the CAD model dataset and the NX Motion dataset. The NX Motion dataset includes the NX Motion simulation model and the afu-files appended to the simulation model. These afu-files are listed in Table 5-3 Independent Rigid Body Simulation Models of Case . Result files of a subsystem MBS simulation solve run are also stored in the appended NX Motion dataset of the CAD item revision. Figure 5-28 illustrates the architecture.



**Figure 5-28 Rigid Body Architecture of Case 3**

## II. Flexible Body Simulation Models

One part of the hinge subsystem has to be a flexible body (see Section 5.4.4 III). So, a finite element simulation model has to be generated. The source is the CAD item revision including the dataset with the part that has to be represented as a flexible body. This CAD item revision is subordinated under the CAD item revision of the hinge subsystem including the BOM line. Based on the CAD model part file, an idealised part file is generated and managed as dataset under a new CAEGeometry item revision. The CAEGeometry item revision receives a CAESource and CAETarget relationship pointing to the source CAD item revision (Siemens Product Lifecycle Management Software Inc, 2012). The idealised part file is the source for the dependent mesh-file which is managed as dataset under a new CAEModel item revision. In order to link the new CAEModel item revision to the appended CAEGeometry item revision and CAD item revision, the CAEModel item revision receives a CAESource relationship pointing to the appended CAEGeometry item revision and a CAETarget relationship pointing to the appended CAD item revision (Siemens Product Lifecycle Management Software Inc, 2012). Based on the mesh file, a simulation file is generated and managed as a dataset under a new CAEAnalysis item revision. In order to represent the dependency of the depending mesh file, the CAEAnalysis item revision receives a CAEDefining relation pointing to the appended CAEModel item revision. This simulation is built upon a solution to generate a reduced flexible body of the hinge subsystem part. The solve run produces a file including the reduced matrix stored in the rfi-format. The result files generated by the solve run are managed in datasets under the appended CAEAnalysis item revision. The post-process results do not produce additional files.

The item revisions, the relationships and the datasets are shown in Figure 5-29.

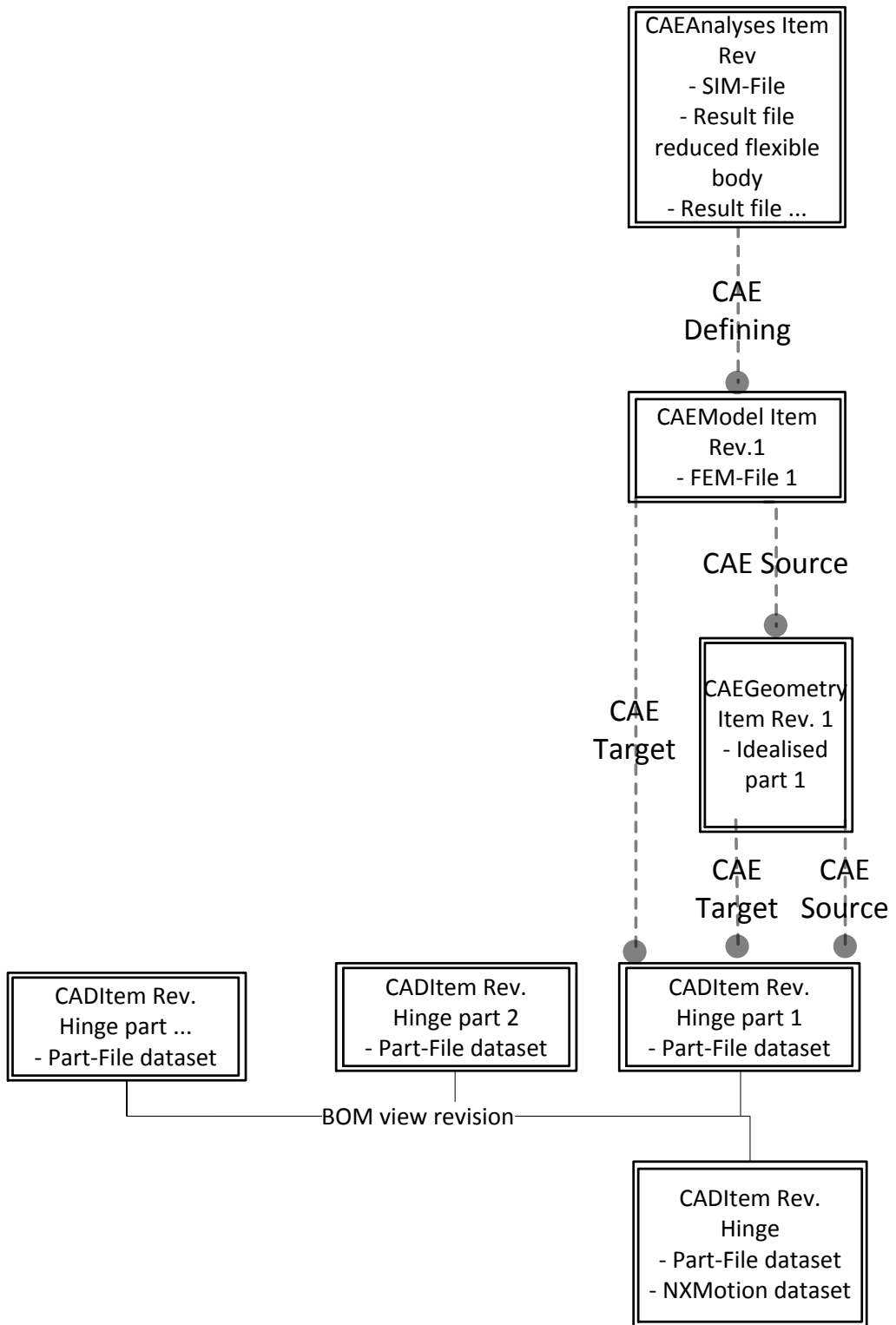


Figure 5-29 Metadata Architecture of the Flexible Body in Case 3

### III. Logical Simulation Models

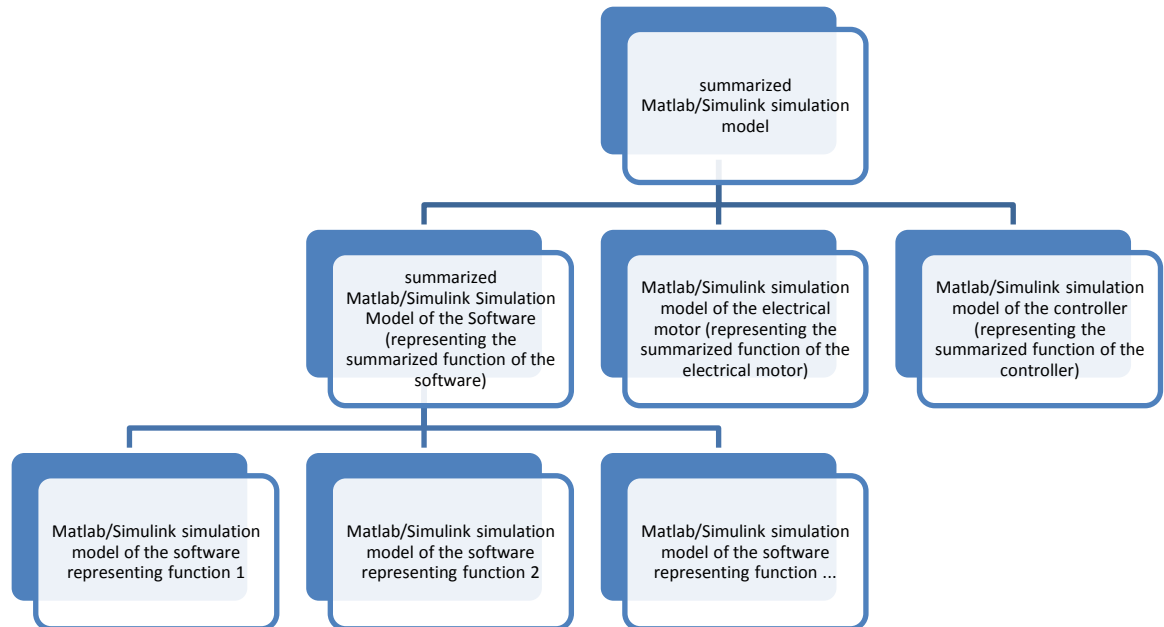
The simulation model of the logical simulation discipline is provided by the customer. This simulation model is a monolithic simulation model comprising multiple subsystems including the controller, the software and the electrical engine (motor) subsystem. The supplied simulation model is generated using the Matlab/Simulink simulation authoring tool. Nevertheless, the supplied simulation model does not mirror a resultant simulation model of the real development process. The customer develops different subsystems from individual development departments (Link, 2012) which causes the generation of separate and focused simulation models of each subsystem. So, each subsystem simulation of the logical simulation discipline is generated in reality, individually and is not assembled.

It is possible to assemble these simulation models manually into one simulation model. As a result, the simulation models have to be generated using the same simulation authoring tool. However, this requires the simulation authoring tool functionality for assembling or connecting different simulation models provided by the Matlab/Simulink simulation authoring tool (MathWorks, Inc., 2011).

For the system-simulation, the monolithic logical simulation model is re-used. However, this does not represent the requirements of real data management where the logical simulation model consists of multiple subsystem simulation models. In order to integrate the generation process of such a logical simulation model, an empirical development process of the logical simulation model is used for the case study. This empirical development process generates the logical simulation model based on three subsystems: controller, software and electrical motor. These subsystems can be provided by a supplier. The idea is that the supplier also provides the simulation model of its system such as an electrical motor subsystem and the controller subsystem. Hence, the suppliers deliver the product descriptions and a Matlab/Simulink simulation model representing their subsystem. This is similar to the vision discussed in Link (2012).

The responsibility for the software subsystem can be partitioned. In the case of software development, the logical simulation model is generated before software code generation and detailing. The challenge is to derive the software source code from the logical simulation model. Such a function is provided, for example, by the simulation authoring tool Matlab/Simulink (MathWorks, Inc., 2012). This saves time and reduces errors. However, the software code of a subsystem is complex. In order to keep the software as simple as possible, the software subsystem is sub-ordered to multiple subsystems. A function model helps to do this beforehand and supports the work on the subsystems by keeping the system and subsystem easily viewable. In Case 3, the functional ordering of the controller software can include the functionality of the electrical engine, the controller and the software, whereby, the software functionality suborders multiple sub functions of the software. The hierarchical ordering of the Matlab/Simulink simulation models can correspond to the function model order. This means that the top simulation model integrates the sub-assembled simulation

models. This empirical ordering of functions or Matlab/Simulink simulation models is shown in Figure 5-30.



**Figure 5-30 Empirically Logical Matlab/Simulink Ordering**

This empirical structure and file management can be achieved with TEAMCENTER behaviour models (see Section 4.8). So, the seven Matlab/Simulink simulation model files are stored as datasets under a behaviour model item revision. Each of the Matlab/Simulink simulation files are modelled with the required input and output ports. The ports are stored as behaviour model input ports or behaviour model output ports, in TEAMCENTER. The connection of ports in a Matlab/Simulink is managed as a behavioural model connection. The port representations and connections help to support the data flow between the Matlab/Simulink simulation models. In addition to port management, the hierarchical ordering of the behaviour models is managed using BOM view technology. The hierarchical structure represents the Matlab/Simulink simulation model hierarchy shown in Figure 5-30. This empirical structure of the logical simulation and its dependencies are shown in Figure 5-31. The empirical view changes the single summary simulation model of the logical simulation to a summary of multiple logical simulation models which correspond. The corresponding top simulation model is organised by a top-logical simulation model. These Matlab/Simulink models and files are organised and ordered by BOM view technology.

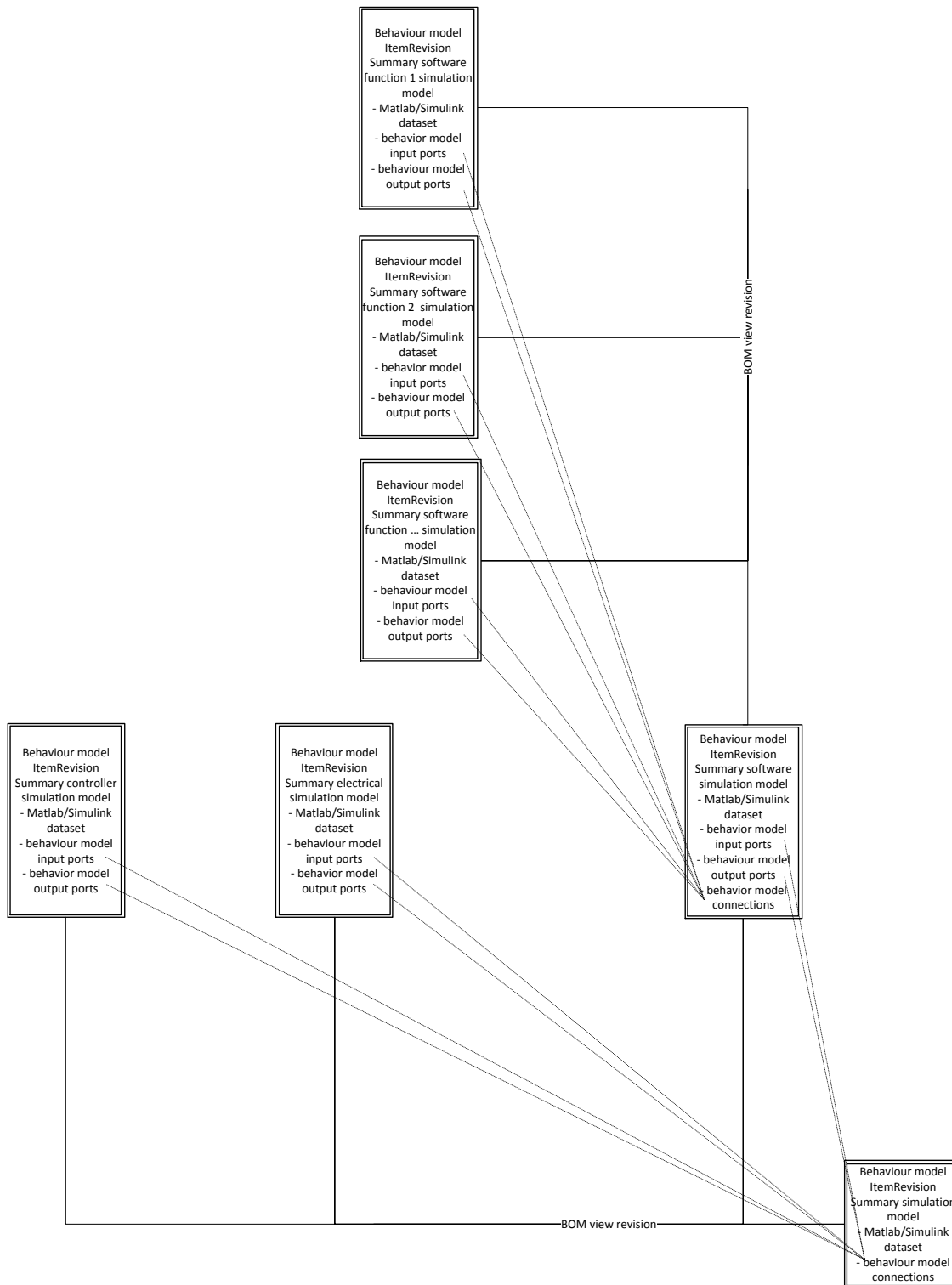


Figure 5-31 Behaviour Model Structure of Case 3

The logical subsystem can achieve a complex hierarchy. The empirical data of this case example is kept simple and understandable, but the reality can be much more complicated. The organization of the hierarchical ordering of logical models is supported by a previously generated functional hierarchy.

The lift gate system simulation requires logical simulation and additional simulation models of other disciplines. The following will explain the assembly of multiple simulation disciplines.

#### IV. Summarised Simulation Architecture

The lift gate simulation requires a combination of the logical, flexible and rigid body simulation disciplines. As discussed before, the simulation disciplines, generated models and data management were discussed. In order to generate the lift gate system simulation, these three simulation disciplines and their simulation models have to be collected, assembled and connected. Data management should support the collection of the simulation models. Afterwards, the collected simulation models are provided for system-simulation pre-processing. This means that the simulation models are assembled and connected to prepare the system-simulation solve run. NX motion provides the functionality to assemble and connect different simulation disciplines. This achieves a co-simulation of the three simulation disciplines. NX motion is also used to model the simulation models of the rigid body simulation discipline.

The interaction of the flexible body with the rigid body is achieved by the exchange of a rigid body link via a flexible body description. So, the rigid body link of a part in the hinge subsystem is exchangeable with a pre-generated flexible body. The process and the required rfi-file were discussed in Section 5.4.4 II: Flexible Body Simulation Models and III: Architecture of the Flexible Body Simulation Discipline of Case 3. The connection points of the rigid body link and the flexible body are congruent. Using these congruent points provides the possibility of exchanging the deformation and force parameters of the congruent points between the flexible and rigid body simulation disciplines and solvers.

The data management structure to achieve the co-simulation interaction of the rigid and flexible simulation discipline is shown in Figure 5-32. This figure combines Figure 5-28 and Figure 5-29. The combination of both enables the re-use of the rfi-file of the flexible body generation simulation. This rfi-file is the reduced matrix representing the reduced flexible body of the hinge subsystem part. By exchanging the rigid body link, which represents this hinge subsystem part in the rigid body simulation, achieves the co-simulation of the rigid body simulation solver and model with the flexible body simulation solver and model. The integration of the reduced flexible body matrix file (rfi-file) is not provided automatically by TEAMCENTER. The reduced flexible body matrix file has to be copied manually into the NX Motion dataset of the lift gate system CAD item revision. The copying causes duplication and



loses the dependency between the copied and the original source. If there is a change to the original source, the copied source will not be changed and there will be no dependency to check the veracity of the copied source.

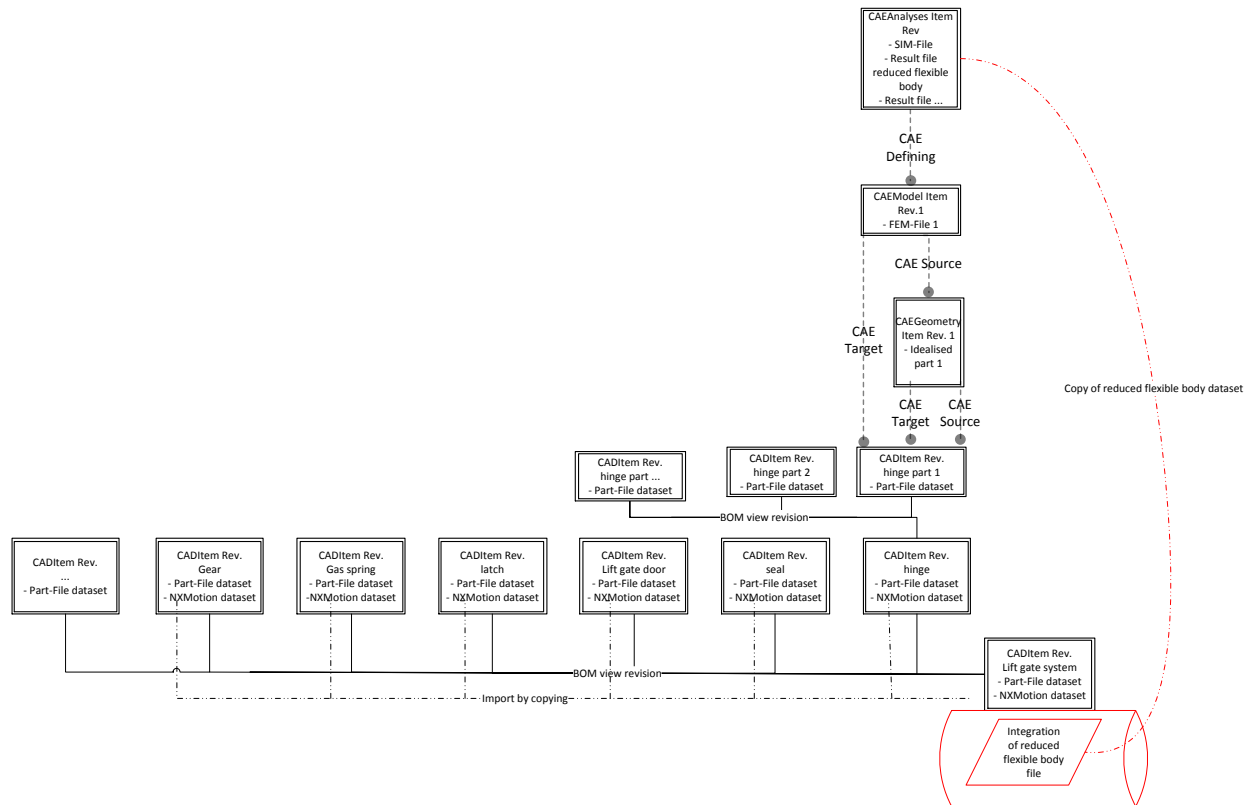
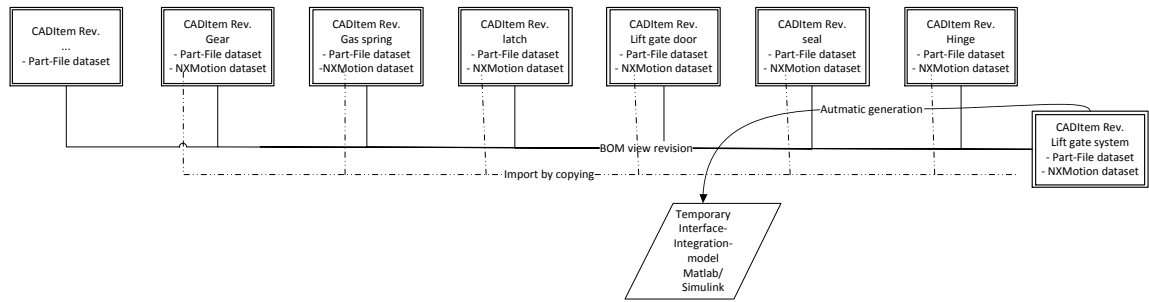


Figure 5-32 Integration of the Flexible Body

Furthermore, the co-simulation of the rigid body and logical simulation discipline can be defined with NX Motion. Siemens offers an add-on tool with NX Motion Control that provides the capability of interaction between the NX Motion solver and Matlab/Simulink. So, Matlab/Simulink and NX Motion has to be configured. This configuration is mainly carried out in NX Motion where the input and output parameters to the Matlab/Simulink model are defined. NX Motion generates a temporary interface integration model for Matlab/Simulink including an s-function block. This s-function block, provided in the temporary interface integration model, includes a configuration for the solver interaction and can be re-used in a pre-generated Matlab/Simulink simulation model. Via drag and drop, the s-function block can be re-used and integrated into a Matlab/Simulink simulation model that should be coupled. This is shown in Figure 5-33. The temporary interface integration model is not saved because it is only required for the drag and drop procedure into the pre-generated Matlab/Simulink model.



**Figure 5-33 Temporary Interface Integration Model for Matlab/Simulink**

The s-function block from the temporary interface integration model has to be assimilated into the Matlab/Simulink model of the summary simulation model. The integration of the s-function block into the summary simulation model file is achieved in the co-simulation of the rigid body simulation and the logical simulation. The summary simulation model is a single file. However, the empirical viewpoint (see Section 5.4.4 III) of the logical simulation model brings the summary simulation model into a simulation model that consists of subordered simulation models. The s-function block from the temporary interface integration model is integrated per drag and drop into this logical simulation model. However, this coupling is not uniquely reviewable in the data management tool. The dataset under the behaviour model item revision which can be used for the logical simulation data management is not uniquely identifiable. Additionally, there is no automatic provision for subordinated behaviour model item revisions at a top level behaviour model item revision. Manual copying of the logical simulation data to the NX Motion dataset of the CAD item revision is the easiest way to provide the required data. However, this causes the loss of a dependency between the copied and original source. If there is a change to the original source, the copied source will not be changed and there will be no dependency to check the veracity of the copied source. This is shown in Figure 5-34.

The collection of the simulation data and the assembly and connection of different simulation models was discussed earlier. The system-simulation combines the simulation of multiple solvers with a co-simulation. The NX Motion and the Matlab/Simulink solvers are directly coupled. The flexible body simulation discipline was archived by a reduced matrix of the finite elements representing the flexible body. The reduced flexible body is solved directly and integrated into the NX Motion solver. Flexible body results of a few predefined points are generated. In order to generate all results of the flexible body simulation model, a down-streamed finite element solve run is carried out. In the case of NX Motion and NX Motion Flexible Body, this finite element solve run occurs automatically because the results of the finite element solve run can directly be post-viewed with NX Motion Flexible Body.

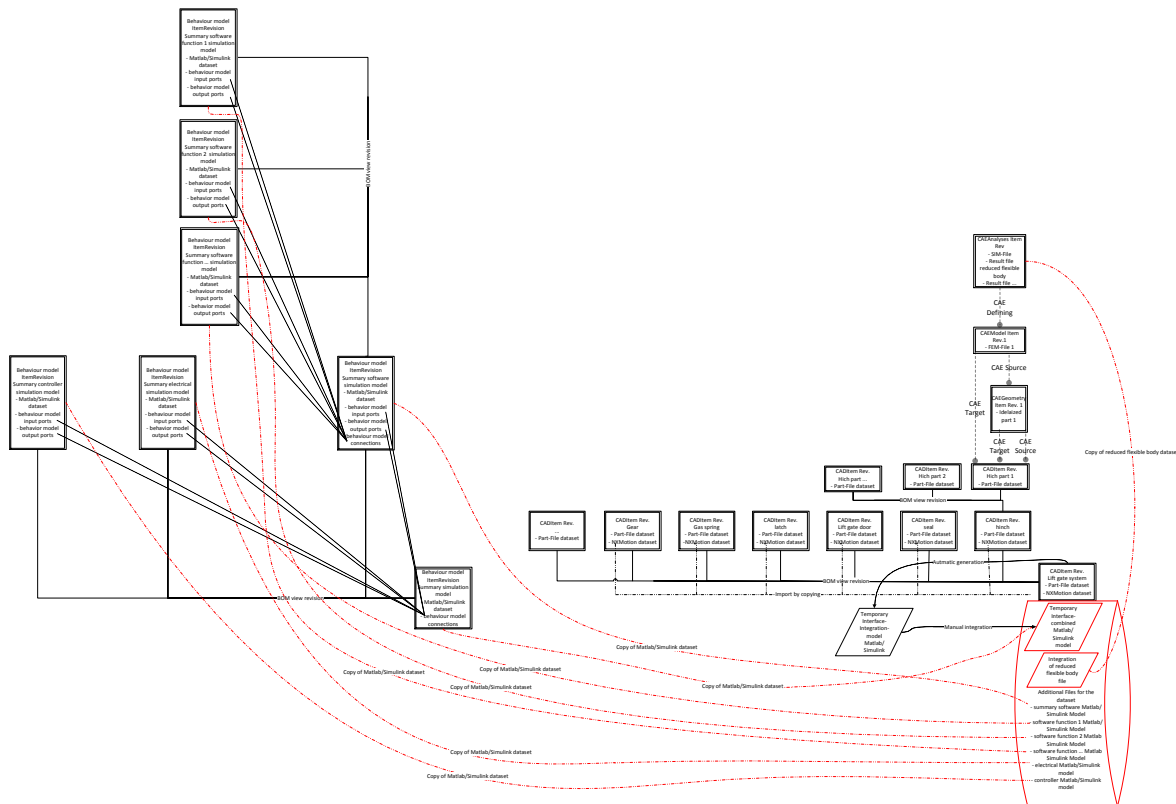


Figure 5-34 Summary Data Management of Case 3

All the result files are managed under the NX Motion dataset of the top CAD item revision where the top rigid body simulation is also managed. However, the result files are dedicated to the special simulation models and managing item revisions:

- The result files of the flexible body belong to the CAE Analysis item revision.
- The result files of the rigid body belong to the CAD item revision.
- The result files of the logical simulation belong to the summary simulation behaviour model item revision.

The management of the results under the CAD item revision dataset does not reflect a realistic dependency. Traceability is not provided this clearly. Due to the loss of the system-simulations result traceability, it is not possible to load the pre-process data and results for post-processing. This does not appear to be an issue in this case; but in other application cases, the missing functionality of supporting the post-process can cause extensive issues for other departments and experts. Consequently, manual data handling is required in this case.

For this case, the data management of all the required data has been achieved. However, unique traceability is not achievable due to some missing, unique dependencies. TEAMCENTER offers additional possibilities that can reduce these issues. Such a possibility is customization. However, a customization of TEAMCENTER is a specialization and does not

provide a general solution. Additional customizations can reduce flexibility. The customization requires special resources such as administrative TEAMCENTER specialists. These specialists are not simulation specialists. This makes the organization and the management of the customization project difficult and causes the configuration to provide a solution for recurring simulation works to have a high rate of recurrence. However, configurations are not a solution for simulation work with low recurrence rates.

The new framework for the support and management of multi-disciplinary simulation data embedded in a PLM environment should improve this situation. In Section 5.4.5, Case 3 will be empirically integrated into the new framework.

#### 5.4.5 EMPIRICAL VALIDATION OF THE NEW FRAMEWORK AND CASE 3

The new framework for the support and management of multi-disciplinary simulation data embedded in a PLM environment should improve the actual possibility of multi-disciplinary simulation data management. The files, data and information should be uniquely dependent and traceable. So, two new structures will be integrated:

- 1) The system simulation structure (System simulation item revision) to manage the multi-disciplinary system simulation structure. This structure uses BOM view technology to suborder the required simulation model item revisions. The simulation discipline or tool of the simulation models is independent of the sub-ordering of the simulation model item revisions. In Case 3, the sub-ordered simulation models will be
  - a. the item revisions of the flexible body simulation model that merged the sub-ordered rigid body systems,
  - b. the rigid body simulation model, including the rfi-file, and
  - c. the logical simulation model.
- 2) The second added structure of the new framework is the system simulation result structure to manage the result files, data and information produced by the system simulation solve run. This structure is based on System Simulation Result item revisions. These System Simulation Result item revisions can be generated for each simulation model that takes part in the system simulation solve run. BOM view technology helps to order the System Simulation Result item revisions. The files, data and information generated by the solve runs of the single simulation models are managed as datasets under the System Simulation Result item revisions.

These two new items require a dependency representation. The source system-simulation relationship points to the source item revision of the (system-) simulation item revision. The target system-simulation relationship points to an item revision that is represented by the

(system-) simulation. So, the sources and represented items of a simulation item, which can take part in a system-simulation or be a system-simulation on its own, are uniquely traceable. In Case 3, there are three source and target system-simulation relationships:

- 1) The logical (top) behaviour model is based on the top functional description. This is represented by a system-simulation source relationship pointing from the behaviour model item revision to the functional item revision. Additionally, the system-simulation target relationship with the same linking objects and direction denotes that the behaviour simulation model represents the top function.
- 2) The flexible body simulation is based on the hinge subsystem. This is represented by a system-simulation source relationship pointing from the CAEAnalysis item revision to the CAD item revision. Additionally, the system-simulation target relationship with the same linking objects and direction denotes the flexible body simulation model which represents the hinge subsystem.
- 3) The rigid body simulation model is based on the CAD models. This is represented by a system simulation source relationship pointing from the CAD item revision to the CAD item revision. Additionally, the system-simulation target relationship with the same linking objects and direction denotes the rigid body simulation model that represents the summarised CAD assembly. In this case, the rigid body simulation model is managed as dataset under the CAD item revision. Optionally, the rigid body simulation model can also be managed as a dataset under a NX Motion item revision which can achieve an enhanced overview of the data management solution.

The second new item is the system simulation result item revision. This item is linked by a CAE Result relationship pointing from the system simulation result item revision to the item revisions of the result generation responsible simulation models. These result generation responsible simulation models are the rigid body dataset holding item revision, the flexible body dataset holding CAEAnalysis item revision and the behaviour simulation model item revision. Based on these relationships, the existing results of the system-simulation and sub-ordered simulation models can be identified.

The new framework provides a superior and clearly organised data management of the data, files and information taking part in the system simulation of Case 3. Functionalities, included in the new framework, can now provide the required data, files and information for the simulation process. It has to be understood that the validation of the new framework with the Case 3 is empirical.

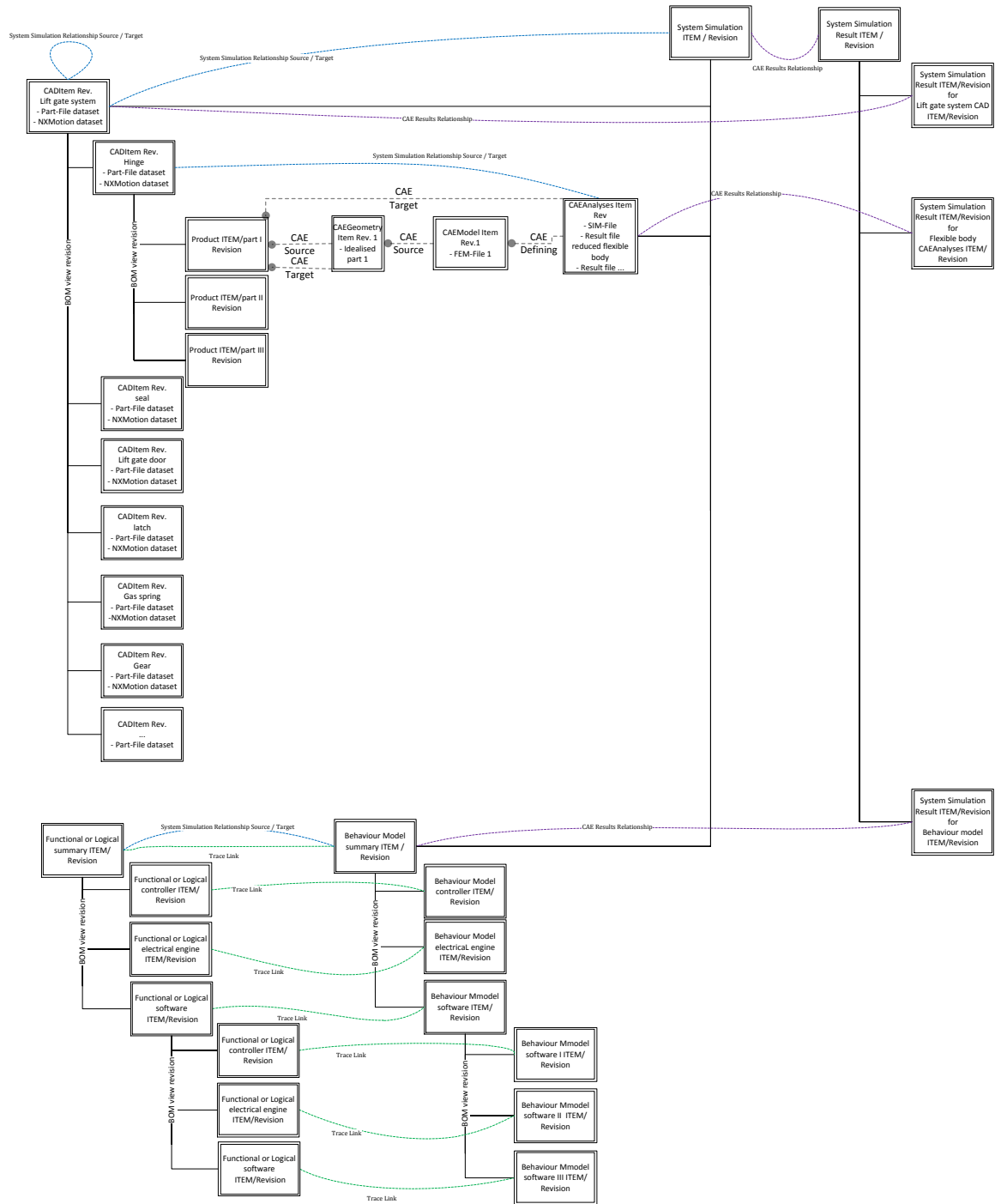


Figure 5-35 Empirical Structure of Case 3 in the New Framework

System simulation can also be seen dependent upon a model-based development process. This will be discussed in the next case example and section.

## 5.5 CASE 4: PROJECT: “INTERDISCIPLINARY MODEL-BASED DEVELOPMENT PROCESS”

The development of a complex system can be improved by integrating a system-simulation of this system into an early development stage. This is achieved using the model-based development methodology. The concept of the system should be optimised by an early system-simulation. Afterwards, the discipline-oriented development departments start the detailing of the system. An automotive company in the south of Germany invests in the integration of this methodology. For the system simulation, the different simulation tools and different physical disciplines have to interact with each other. A research project should help to verify and validate existing commercial tools from the view point of interdisciplinary model-based development methodology. However, the goal is not to identify the best tool. The focus is on the improvement of simulation software tools in response to the development methodology in order to create an optimal and generic interdisciplinary model-based development process. So, the electrical front door windows lifter system of the automotive company was used as the case study.

### 5.5.1 DISCUSSION OF THE ELECTRICAL FRONT DOOR WINDOWS LIFTER SYSTEM

In the “Interdisciplinary Model-Based Development Process” project, an improved development process should be developed. The electrical front door windows lifter system was used as a case example. This electrical front door windows lifter system is the product of multi-disciplinary and inter-disciplinary subsystems. The following describes the subsystems:

- The electrical engine subsystem creates torque and drilling movement.
- The gear subsystem transforms the drilling movement and torque into lower drilling speed and higher torque.
- The cable pull subsystem transforms the high torque and lower drilling speed into a three dimensional translator force and movement. The cable pull also includes a damping effect on the movement of the windows lifter system.
- The glass panel subsystem protects the passenger. It includes protection against forces from air or crash, as well as protection against water and temperature. The glass panel should also optimise driving resistance to reduce driving energy.
- The seal subsystem protects the passenger. It includes the protection of the glass panel. The seals also provide a flexible link between the glass panel and the chassis. The material properties and the movement of the glass panel require a flexible link between chassis parts and glass panel.
- The chassis subsystem provides a framework where the parts can be positioned and fixed.



- The controller subsystem manages and controls the movement of the glass panel. This includes the protection of humans in the case of moving glass panels.

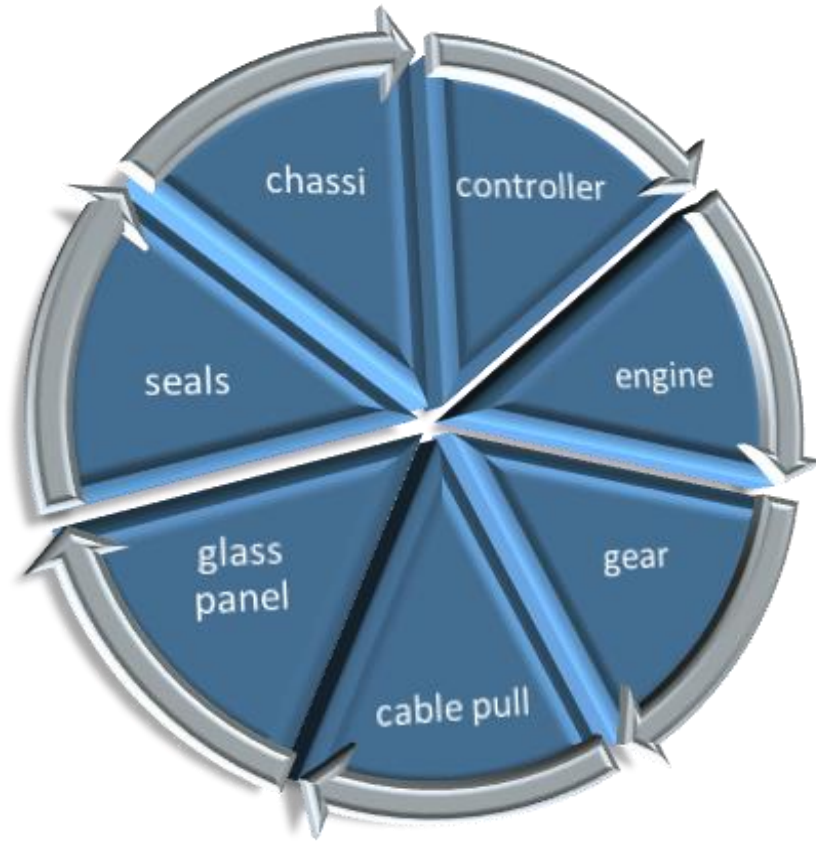


Figure 5-36 Electrical Front Door Window Lifter System

Mahler(2012) discussed and presented the system:

“The controller will influence the engine and get movement information back from the engine. Based on the data from the engine the controller will calculate the electrical input for the engine. The engine will move the gear. Caused by the movement the gear will react with resistance. This resistance will be generated by the movement of the cable pull, glass panel and seals. The cable pull is a complex system [...] extremely complex in mechanics. Springs will strain the cable. There is a lot of friction caused by the guide of the cable. The cable on its own will be also a spring system. These spring- and friction-factors will cause a spring-damper-system. This spring-damper-system has a great influence on the action-reaction between gear and glass panel. Controlling input will neither be proportional nor similar to the glass panel movement which should be controlled by the controller. The glass panel will be forced to move by the cable pull. But movement of the glass panel will react to guide and friction forces in the seals. A system like this will be extremely hard to control. Small changes in the system, like lesser friction, will cause a significant change in the controller.”

This system and subsystem are used as system-simulation cases. Additionally, work results on the interdisciplinary model-based development process of the automotive customer project are integrated into this system-simulation case. In Section 5.5.2, the case example will be specified, and the simulation and the sub-simulations, discussed.

### 5.5.2 DISCUSSION OF THE SYSTEM- AND SUBSYSTEM-SIMULATION OF CASE 4

The system of the electrical front window lifter of a car door is sub-ordered into the following subsystems:

- 1) Seal system
- 2) Glass panel system
- 3) Chassis
- 4) Controller system
- 5) Software system
- 6) Cable pull system
- 7) Gear system
- 8) Electrical engine system

The interdisciplinary model-based development process should help to develop the subsystems of the electrical front window lifter system from concept through to prototype. For the development process, a system-simulation is required. So, the subsystems can be modelled independent of each other. In the following paragraphs, the useable simulation tools and their cooperation will be discussed. The multiple simulation models of the subsystems will be summarised in the simulation architecture.

#### I. Structure of the Simulation Architecture

The structure of the simulation architecture is influenced by the simulation models of the subsystems. Mahler(2012) discussed that:

“Necessary are different mathematical bases to simulate the different subsystems. The mathematical simulation base could be specified in flexible bodies, rigid bodies and logical. [...] The things like controller or software will be logical; they have direct influence on one or more rigid or flexible bodies. [...] Flexible will be the seals because the force caused by the movement of the glass panel in the seals will be dependent on the deformation of the seals. The glass panel deformation could be negligible. The seals will also be forced and deformed by the clamping into the chassis. For simulating this attitude a model including flexible seals, the chassis influencing parts as well the glass panel has to be built up. The flexibility of the seal will be complex. Requirements for the high level of nonlinear material attitude have to be solved by the simulation – tool. This flexible attitude will mostly be caused by nonlinear contact that will be additional a requirement for the solver.”

This means that the seal, glass panel and some chassis parts are treated as flexible body simulations because the flexibility of the bodies is important. On the contrary, the flexibility of the cable pull, gear and electrical engine is negligible and can be treated as rigid body simulation. The controller and the software represent a logical behaviour of the system instead of a body-oriented behaviour; these subsystems are represented as logical models and are illustrated in Figure 5-38.

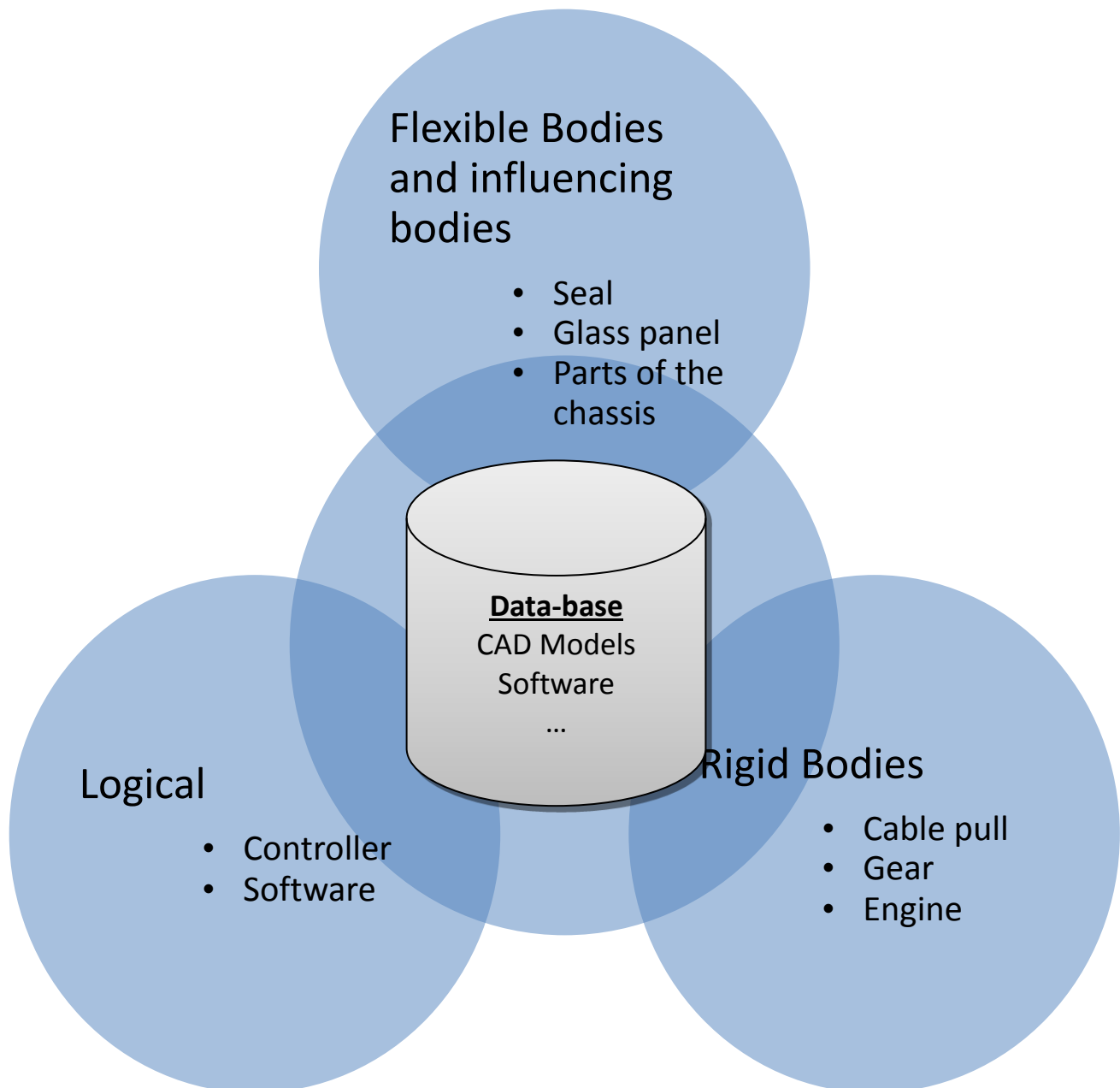


Figure 5-38 Case 4: Simulation Discipline Architecture (Mahler, 2012)

The three simulation disciplines, flexible body, rigid body and logical simulation, can be achieved with the tools described in Table 5-4 and Figure 5-39.

Table 5-4 Case 4: Table of Simulation Tools

Simulation Discipline	Simulation Tool
Flexible body simulation	NX Advanced Simulation and add on Solver NX Nastran Advanced Nonlinear
Rigid body simulation	NX Motion and maybe add on NX Motion Control
Logical simulation	Matlab and add on Simulink

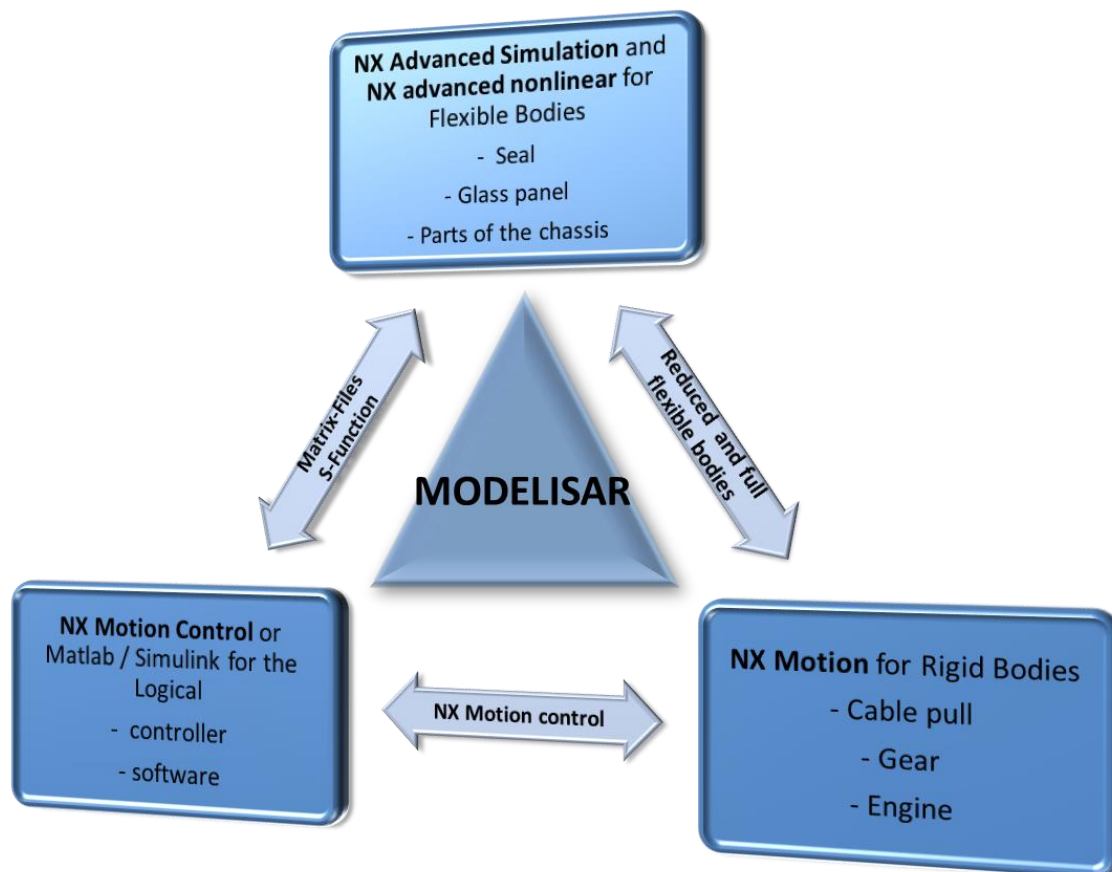


Figure 5-39 Case 4: Simulation Tool Architecture (Mahler, 2012)

For a system-simulation, these simulation tools have to be coupled. The MODELISAR consortium(2010) discussed that the MODELISAR interfaces are able to couple simulation models and solvers vendor-independently on a parameter exchange base. So, the MODELISAR interface makes it possible to couple all the systems of Case 4. In the case of NX Motion and

NX Motion Control, the existing interface to Matlab / Simulink is superior because this technology can be used without configuration of the software. In the case of using MODELISAR functional mock-up interface (MODELISAR consortium, 2010), the simulation tool NX Advanced Simulation and the add-on solvers NX Nastran Advanced Nonlinear and NX Motion have to be improved to support the MODELISAR functional mock-up interface (Mahler, 2012). So, this tool set does not support this technology. Nevertheless, the existing coupling technologies of NX Motion are useable.

In the case of the flexible body simulation, this simulation can take a long solve time because the required simulation model is nonlinear. So, an alternative way for a simulation coupling can be of interest. In this case, a field of force, which moves the window, is identified dependent upon the window panel position and the window panel velocity. In order to generate this field, an independent and flexible body simulation has to solve multiple occurrences with different velocity and position parameters of the window panel. The results are useable to work out the field of force, dependent upon the window panel position and the window panel velocity. The moving forces, dependent upon position and velocity, span a response surface. This means that a response surface should be derivable from the flexible body simulation model of the seals system. This can be achieved much more comfortably with the new technology described in Dr. Hartmann & Mahler (2013). Nevertheless, this technology was not given at the working stage of the case example. So, the seals simulation has to run manually and which is extremely time consuming for the user. The response surface has to be generated manually, as well.

A fourth possibility of communication is the use of simulation middleware which provides interfaces to the different simulation solvers. So, simulation solvers can interact with the simulation middleware. Such an interface could be via the MODELISAR FMI/FMU (Functional Mock-up Interface / Functional Mock-up Unite) interface. However, such interfaces are not implemented into NX Motion or NX Advanced Simulation. This means that the simulation tools, NX Advanced Simulation and NX Nastran Advanced Nonlinear solver and NX Motion, have to be modified to support such interfaces to simulation middleware.

The next section explains how the simulation tools should work together in a simulation process.

## II. Co-Simulation and Simulation Process Concept

For a system simulation of the electrical front window lifter of a car door, multiple simulation models and simulation disciplines have to cooperate simultaneously. The necessity of three simulation models was described and explained in Section 5.5.2 I and shown in Figure 5-38 and Figure 5-39. These three simulation models are based on specific simulation disciplines and include:

- a rigid body simulation model for the cable pull, gear and engine; implemented with the NX Motion simulation tool,
- a flexible body simulation model for the seal, glass panel and parts of the chassis; implemented with the NX Advanced Simulation tool and
- a logical simulation model for the controller and software; implemented with Matlab/Simulink.

The rigid and flexible body simulation model would be based on CAD models as well the product structure. However, the logical simulation model is based on the controller and software model. The controller and the software are not described as CAD models. In order to describe controller and software functionalities, an industrial approach is to use functional descriptions and structures. Nevertheless, the three simulation models can be generated individually and independently from each other.

One result is that each of the three simulation models is solvable on its own. This provides multiple possibilities to handle the communication and cooperation of the simulation models for system-simulation generation. There are three possibilities reviewed to couple the simulation tools:

- (1) There is an interface provided with NX Motion control that supports the co-simulation between NX Motion and Matlab/Simulink simulation models. Flexible bodies can also be integrated with NX Flexible Body. This interface supports reduced flexible bodies. In this case, the seals have to be modelled as a nonlinear FEA simulation model. A nonlinear FEA simulation model would not be reducible to a reduced flexible body. This kind of FEA simulation model has to be integrated as a full flexible simulation model. The integration of the nonlinear FEA simulation model as a full flexible body is not supported by NX Motion. Figure 5-40 shows the cooperation between the described interfacing technologies.

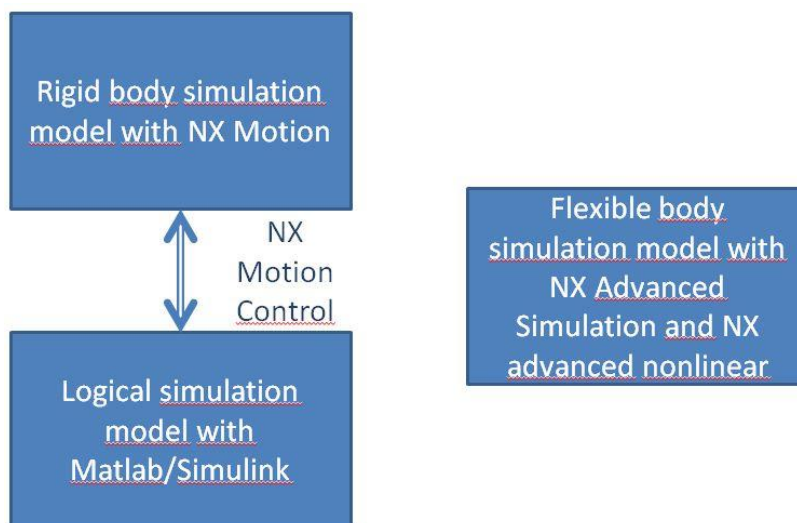


Figure 5-40 Assembling Version1 of Case 4

- (2) There is a possible alternative for the missing interface of the FEA simulation model. The FEA simulation model runs separately with predefined and quantified input parameters. These action input parameters represent the missing interface input parameters, mainly coming from the rigid body simulation model. The reaction parameters, which are the output parameters of the interface, have to be stored dependent upon the input parameters. These action and reaction parameters of the FEA simulation model run can represent the missing FEA simulation interface parameters. So, the reaction parameters are saved dependent upon the action parameters in a parameter file or a field file. Figure 5-41 shows the cooperation between the described technologies.

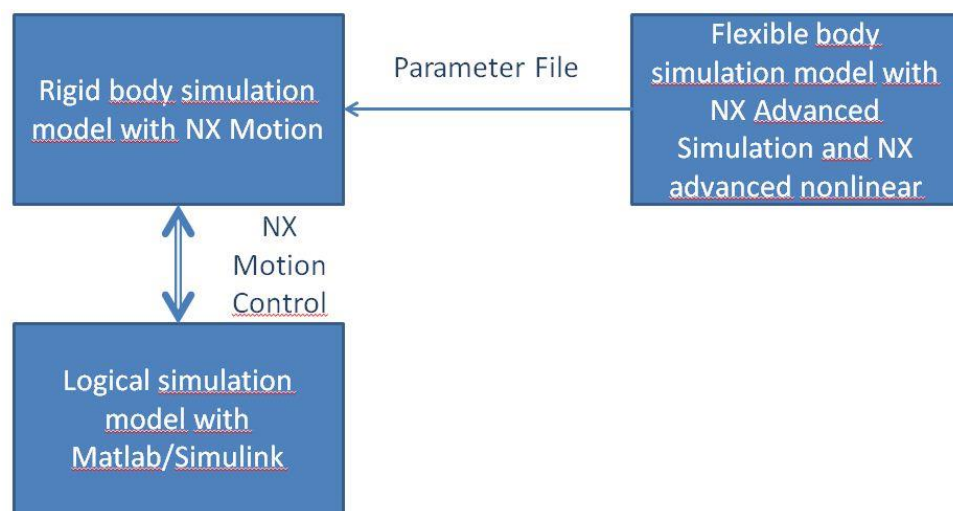


Figure 5-41 Assembling Version 2 of Case 4

- (3) Another solution is to use a neutral interface. The MODELISAR FMI is independent from the simulation software vendors. It supports a framework that can achieve an interaction between all three individual simulation tools via the MODELISAR function mock-up interface. Other tools such as Dymola or Matlab/Simulink can interact with MODELISAR function mock-up interfaces. However, the simulation tools NX Motion and NX Advanced Simulation do not support the MODELISAR FMI and cannot create the MODELISAR FMUs. Nonetheless, the MODELISAR FMUs can also be created manually, the disadvantage being that manual creations require additional work and reduce stability.

As an alternative to the direct interaction of some simulation tools, the use of simulation middleware is possible with the MODELISAR FMI and FMU technology. The simulation middleware allows communication between different running simulation models and solvers. It also synchronises the system-simulation solve run with the sub-ordered simulation models and solvers. This will be shown in Figure 5-42.



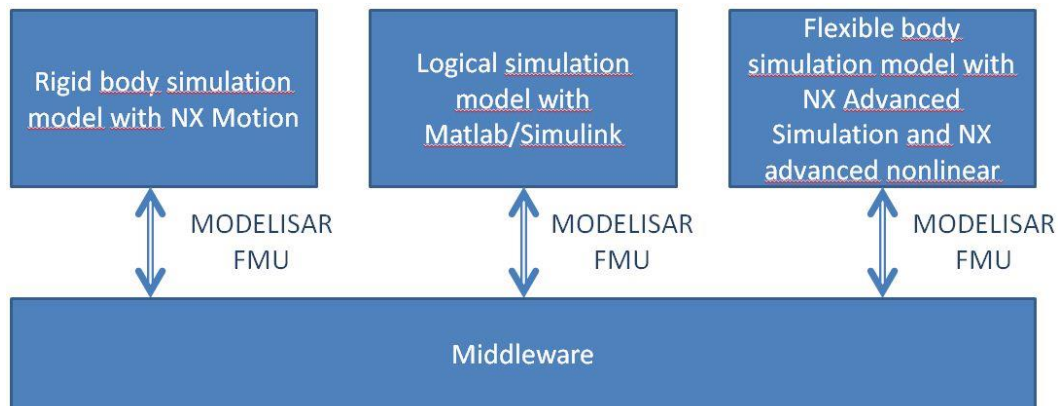


Figure 5-42 Assembling Version3 of Case 4

The aforementioned automotive customer showed an interest in Version 3. With this solution, the solver tools are more flexible which means that simulation software such as Matlab/Simulink can be replaced by another simulation tool. Due to the missing functionality of the NX tools, this communication version was not feasible. A feasible communication version seems to be Version 2 with constraints. The field generation of the fully flexible body simulation was not achieved due to a high investment in solver run times and manual field generation work. Instead, the case example uses a generic field.

Nevertheless, Case 4 should focus on researching the communications within Version 3. This communication version is more flexible and future-oriented. This flexibility is required to support a system-simulation during the development process.

### III. System-Simulation during the Development Process

One result of the project “Interdisciplinary Model-Based Development Process” was that the system-simulation of the electrical front window lifter of a car door is required during the whole development process. The development process is triggered and controlled by milestones. A result of the project revealed that a system-simulation of the electrical front window lifter of a car door could provide an improvement to some milestones. This kind of development methodology can be called “System-Simulation Driven System Engineering”.

For the system-simulation driven system engineering development methodology, system-simulation models are used at different stages of the development process. The first system-simulation is used for the verification and validation of the mechatronic concept of the electrical front window lifter of a car door. Mahler(2012) discussed:

“The mechatronic system concept will be worked by using simulations to optimise the concept. But in the reality, systems with high influence and complexity between the subsystems will be not safe to fail. An improvement of the development methodology will be useful to optimise the development of systems as described before. Those systems will have mostly high subsystem-controller dependences (Dr.-Ing. J., Dipl.-Ing. S., & Prof. Dr.-Ing. H., 2010) where small tolerances in the subsystems could cause system failure. To identify a failure of a system in the case of maximum tolerances will be not integrated in the system-engineering-process. So, the idea is to control the development process by the maturity of the virtual product. This should identify a failure of the system in an early state and reduce failing in a late state and keep the development in time.”

System-simulations are used for maturity control. System-simulations can generate quantitative and qualitative data and information that can be compared to expected data and information. The expected data and information are defined through requirements that are part of the system engineering methodology, and can provide the basis of the maturity calculation.

However, the use of system-simulations during the development process causes changing bases for the single simulation models. The bases for simulation models in an early development stage are rare. Data such as CAD models are not available. Usually, the system-simulation for mechatronic concepts is only based on functional structures describing the functions of subsystems. Logical simulation models at this development stage are ideal. Later, the first but less detailed CAD models are available. Detailed simulation models are counterproductive because their inputs are less detailed. So, there are simplified simulation models based on the less detailed CAD models. For each progressive step in the development process, the available base data for the simulation becomes more detailed. This improved data, generates simulation models of higher precision and quality. These improved simulation models, which represent the same subsystem, reflects the improved quality down through the hierarchy by changing the sub-simulation models of the system-simulation during the development process. This means that the system-simulation changes step-by-step in the development process. New generated subsystem-simulation models based on the improved subsystem replace previous simulation models in a revision of the system-simulation.

However, old subsystem-simulation models are unusable with these improvements. Dependent on the requested system-simulation results, simplified sub-simulation models can improve the system-simulation performance. Often, existing sub-system-simulation models have to be improved and revised. However, they are smarter with less precision than those detailed simulation models with high precision and higher solve times. Improved and more detailed sub-simulation models are not required for all system-simulation results; sometimes the earlier and simplified generations can be more economical. So, it is possible that earlier generated simulation models are overworked based on these improved sub-systems, such as improvement of simulation parameters, which can cause multiple variants of simulation models describing the same subsystem, to pop up during the system development process.

There were not enough project resources to initiate an actual discussion into the case of the electrical front door window lifter system. However, the described boundaries of “System-Simulation Driven System Engineering” can empirically influence Case 4.

In Section 5.5.3, the description of the system-simulation will be detailed. The simulation architecture and system-simulation generation of Case 4 will be discussed.

### *5.5.3 DISCUSSION OF THE SIMULATION ARCHITECTURE AND SYSTEM-SIMULATION GENERATION OF CASE 4*

This section will discuss the simulation architecture and the generation of the system simulation. The first three sections will discuss the data architecture of each simulation discipline used in this case study. Section 5.5.4 will explain the linking of the subsystem-simulation models to the system simulation.

#### I. Logical Simulation Model of Case 4

The subsystems controller, software and electrical motor have to be modelled and solved with the equation based simulation tool Matlab/Simulink. This EBS tool can be supported by an existing interface integrated in NX Motion Control. However, Matlab/Simulink is not the main EBS tool of the automotive company. As an alternative to Matlab/Simulink, the automotive company is endorsing logical simulation models achieved with Modelica simulation authoring tools. Chrisofakis, Junghanns, Kehrer & Rink (2011) discussed:

“Daimler uses Dymola and also SimulationX to edit and process Modelica models. Since Modelica version 3.1 there is full compatibility of the plant models both in Dymola 7.4 as well as in SimulationX 3.4. Models and libraries are stored on hard disk as .mo files. Both tools are able to read these files with no specific modification, i.e., they use exactly the same files for displaying exactly the same structure.”

In order to achieve such an openness of the system-simulation, the simulation tools and the data management have to mirror this kind of openness and support multiple authoring tools.

This openness is also required to support the simulation model change affected by the ongoing system development during the development process. The logical simulation is driven by the idea to optimise, tune, validate and debug the system and its subsystems. The paper by Chrisofakis, Junghanns, Kehrer & Rink (2011):

“...presents technology targeted toward the late stages in the development process, like tuning, validating and debugging the entire controller software in closed loop with

simulated plant models. Virtualizing these later engineering tasks requires plant models with increasingly higher quality [...] and near-production controller software [...].”

The paper believes that the increasing level of simulation model quality at the increasing development process state and the simulation creation is supported by Modelica:

“This short development is partly credited to good properties of the Modelica language, which provides outstanding support for the re-use of component models, mainly by providing powerful means to parameterise models and built-in support for a causal modelling.”

With the Modelica language, it is possible to assemble individual Modelica simulation models. These simulation models can be built by individuals from different functional areas within the company. The assembled Modelica simulation model can include simulation models of different software and controllers. Chrisofakis, Junghanns, Kehrer & Rink (2011) explained this necessity: “Automotive control software for a single ECU typically consists of dozens of software modules, developed independently by a team of developers.”

The MODELISAR FMI and FMU provide an improved openness in simulation model cooperation. Other solutions also achieve similar results for logical simulation models. However, automotive companies “[...] started recently to use the FMI developed within the Modelisar project as an export format for Modelica models. This standard is supported by the latest versions of SimulationX, Dymola, and Silver (Chrisofakis, Junghanns, Kehrer, & Rink, 2011). An empirical and theoretical simulation model structure realizable with the Modelica language is shown in Figure 5-43. Another example based on Modelica is discussed by Commerell, Mammen, Panreck & Haase (2008). A more realistic structure for the electrical front door window lifter system example is not available. Instead, an empirical and theoretical structure is used. This empirical and theoretical structure is based on multiple sub-ordered Modelica simulation models. The sub-ordering represents three levels.

This logical simulation model technology is not only used for the description of controller and software subsystems. In the early development stage, no CAD models are available. So, the system is described by functions and functional structures. These functions and functional structures provide the base for logical simulation models. This was mentioned and discussed previously in Section 5.5.2 III. This means that logical simulation models can be and are used to describe the behaviour of other disciplines in a system-simulation approach such as mechanical or thermo-mechanical disciplines (especially at an early stage of the development). In a later development stage, CAD models appear in the development process. These CAD models provide a base for rigid and flexible body simulations. In the following, the rigid body simulation of the electrical front door window lifter system will be discussed.

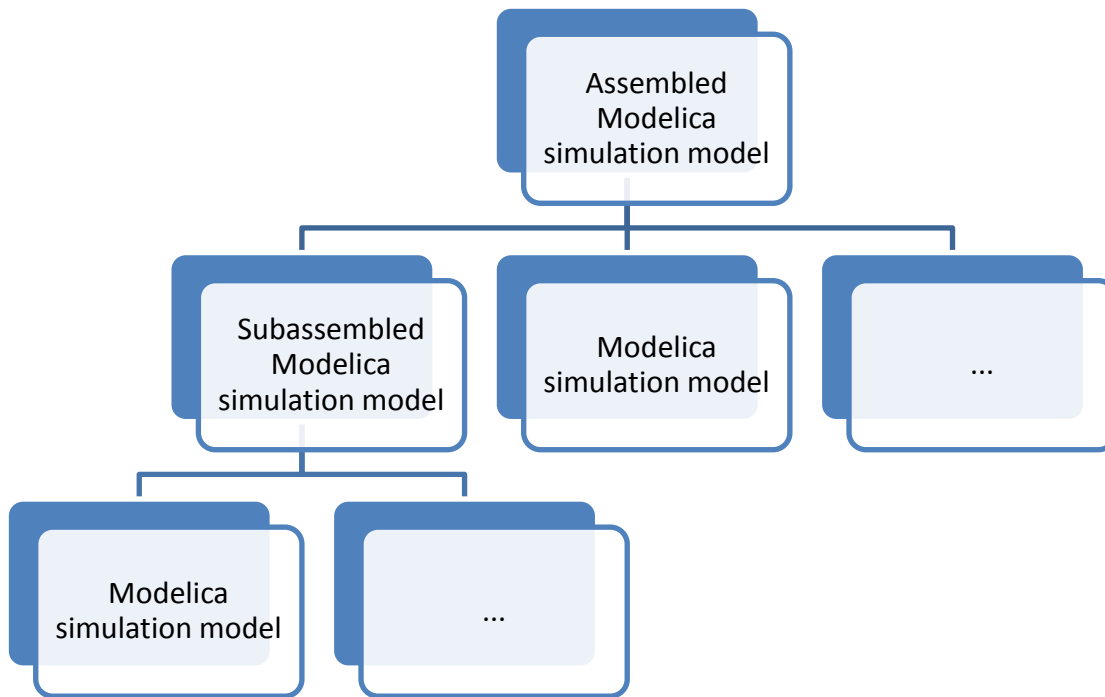


Figure 5-43 Theoretical Modelica Simulation Model Structure of Case 4

## II. Rigid Body Simulation Model of Case 4

The rigid body simulation discipline is accomplished using the MBS simulation tool NX Motion. Everything belonging to the rigid body simulation discipline is integrated into a monolithic simulation model. This simulation is based on an assembly, including all required parts and subassemblies. Included in this simulation model is the cable pull, gear and electrical engine, and the window panel as a moving object (as shown in Figure 5-39). A detailed discussion about simulation generation is not required for the data management case study.

A particular difficulty relating to this simulation model was the modelling of the cable pull. The cable pull is a complex subsystem on its own. Many parameters for describing the effects of damping were missing. Investment is required to generate this missing information. Alternatively, the simulation of the cable pull can also be realised with EBSlike, described in Mammen (2012). In this presentation, the creation of a Modelica simulation model based on the Dymola simulation tool is discussed. An example of such a cable pull is illustrated in Figure 5-44 Example of a Cable Pull .



Figure 5-44 Example of a Cable Pull (dajo Autoteile, 2013)

This system is not developed by the automotive company; it is a supplier part. The idea of the automotive company is to integrate a black-box simulation of the customer instead of creating their own simulation model. However, the supplier keeps the parameters secret. This can be achieved with a Modelica simulation model that is compiled. The compiling of Modelica simulation models is possible because they are c-source-codes and can be compiled in the same way as normal c-source-codes. This compiled c-source-code of the simulation can then interact with other simulation tools using the MODELISAR FMU/FMI technology. The idea of black-box simulations is also discussed by Nybacka, Törlind, Larsson& Johanson (2006). In addition, a black-box simulation was not available for Case 4. Nevertheless, the example depicts the requirement to support interaction with supplied simulation models using the MODELISAR technology.

Instead of rigid body simulation, where deformation of the mechanical parts is negligible, flexible body simulation is required for the system simulation of the electrical front door window lifter system. The flexible body simulation will be discussed next.

### III. Flexible Body Simulation Model of Case 4

The seals consist of hyper-elastic material which provides the possibility of high elastic deformation. In order to represent the self-deformation of the seals, they have to be simulated as flexible bodies. The seal CAD models describe the seal geometry in an undeformed and force-less state. This geometry represents the production geometry of the seals. However, the seals are constrained to the chassis door. Additionally, the montage of the

window panel deforms the hyper-elastic seals. The foundation state of the seals appears in the case of the installed window panel. A pre-simulation is required to work out the foundation state of the seals. After the pre-simulation, the moving of the glass window panel along the seals can be simulated. This simulation has to be based on the pre-simulation because the start of the simulation has to be the foundation of the seals. This kind of simulation is realizable with the NX Nastran advanced non-linear solver.

In order to generate easier manageable meshes, the functionality mesh assemblies in NX Advanced Simulation are used. This functionality provides an opportunity to create independent meshes for a part or an assembly in a product structure. Each mesh is associated and dependent on the source part or assembly. In a higher level assembly, a mesh assembly can be generated. This mesh assembly can identify a mesh or mesh assembly dependent on a part or assembly at a lower level in the product structure. The identified mesh can be connected to the mesh assembly. This assembling also positions the meshes dependent on the product structure, positioned source parts or assemblies (Siemens Product Lifecycle Management Software Inc., 2011). Four seals are required for the simulation of the seals and the windows panel:

- One seal is positioned in the front of the door (red line in Figure 5-45).
- One seal is positioned in the bag of the door (magenta line in Figure 5-45).
- Two seals are positioned in the middle of the door (green and yellow line in Figure 5-45)

CAD models provide the base to mesh the seals. CAD geometry has to be idealised for meshing. So, NX Advanced Simulation provides a file format called idealised geometry (see Section 5.1). Based on the idealised geometry, the mesh is generated and stored in a mesh file. Based on the mesh, the boundary conditions, forces and solver parameters are applied in the simulation file (Siemens Product Lifecycle Management Software Inc., 2011). This simulation model structure is shown in Figure 5-46.





Figure 5-45 Seals in Case 4

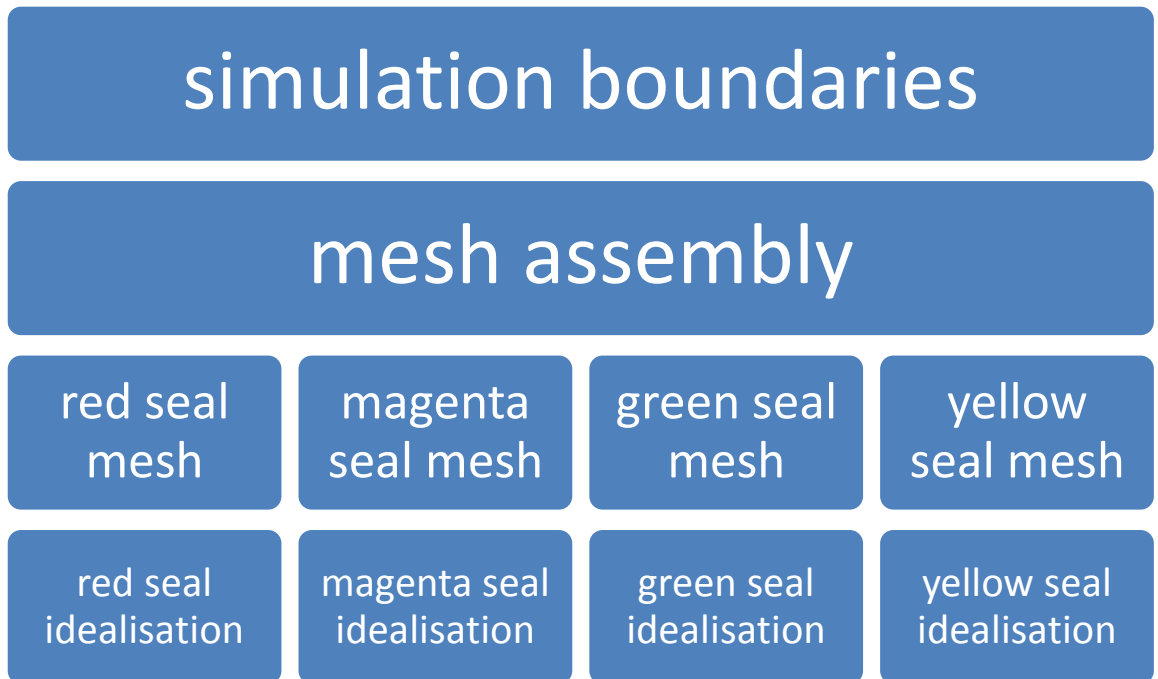


Figure 5-46 Flexible Body Simulation Model of Case 4

In order to reach the basic form of the seals, a first nonlinear simulation has to be created. Here, the window glass panel is positioned dependent on the seal contact forces, and the seals are deformed until they are in contact with the glass panel. Both the window glass panel and the seal's contact forces, have to be in balance. In order to use the result in the next simulation solve run, the solver has to generate a restart file in addition to the result files. The second simulation and solve run is concerned with the movement of the window glass panel in the seals. Movement is defined through velocity and a movement way. The velocity and the movement way should be provided by the rigid body simulation. In future, this data and information should be provided through interfaces from external solve runs in a co-simulation. Currently, this data and information is provided by a parameter or field file. This compromise is required because co-simulation is unprofitable.

The input for the window panel moving simulation is the input deck and the restart file of the foundation seals form simulation. Based on the restart file of the foundation seals form simulation, the window panel moving simulation is done. For easier handling, the window panel moving simulation is split into two sections: the opening procedure simulation and the closing procedure simulation. As described earlier, the opening procedure simulation is based on an input deck and a restart file generated by the foundation seals form simulation. Similarly, the closing procedure simulation is based on an input deck and a result file generated by the opening procedure simulation. These restart files are also copied and renamed. The process is shown in Figure 5-47.

By using the restart files, it is possible to restart multiple times. This provides the possibility of restarting the opening procedure of the window glass panel using different movement velocities. So, the required force to move the glass panel dependent on position and velocity can be calculated. This is the base to generate a response surface or response curve. The data is useable as input for the rigid body simulation.

The flexible body simulation is a subsystem-simulation of the system-simulation. So, the subsystems have to be linked to a system-simulation.

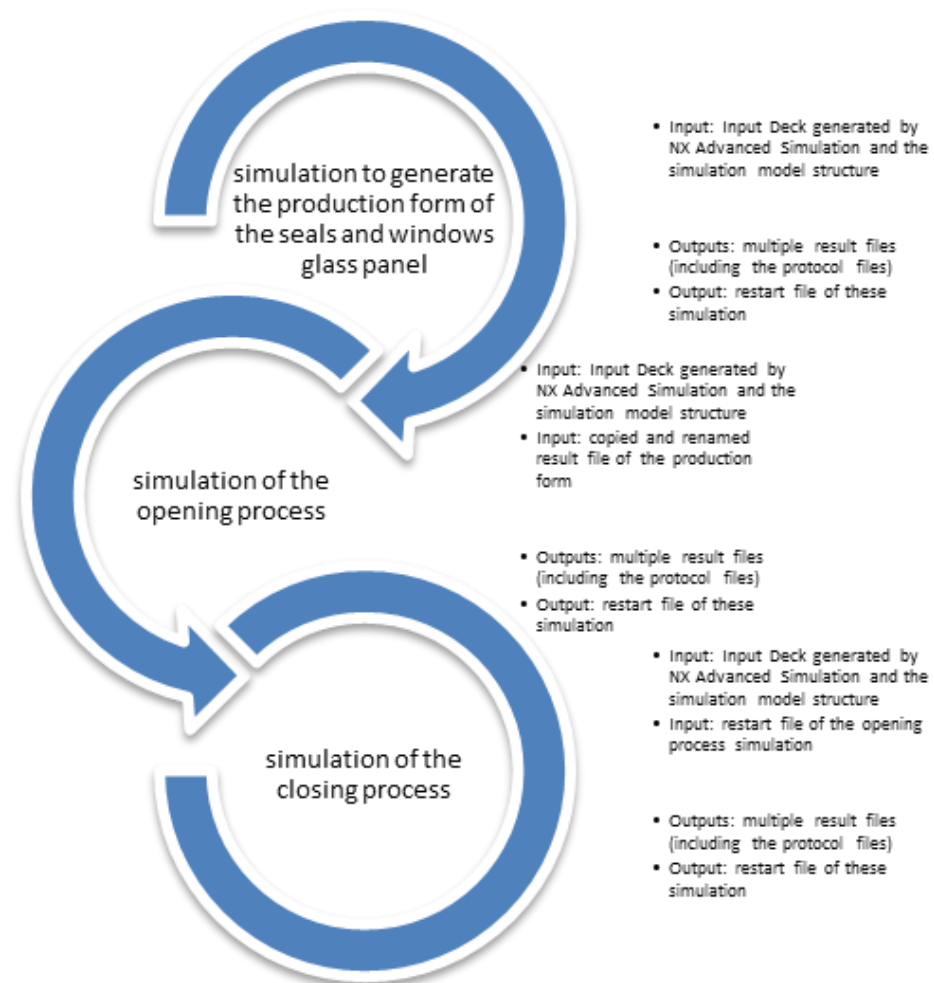


Figure 5-47 Flexible Body Simulation Process of Case 4

#### IV. Assembling of the Simulation Models of Case 4

The assembling and co-simulation of the simulation models should be based on MODELISAR interfaces. The MODELISAR consortium (2010) discussed:

“FMI for Co-Simulation is designed both for the coupling of simulation tools (simulator coupling, tool coupling), and coupling with subsystem models, which have been exported by their simulators together with its solvers as runnable code[...] FMI for Co-Simulation defines interface routines for the communication between a master and individual simulation tools (slaves) in a co-simulation environment. A simulation tool or the part of it prepared for co-simulation by implementing the FMI is called an FMU (Functional Mock-up Unit)”

Two types of MODELISAR functional mock-up interfaces are proposed on the market:

- MODELISAR FMI for model exchange discussed in MODELISAR consortium (January 26, 2010):

“[...] specifies a standardised Functional Mock-up Interface (FMI) for the coupling of two or more simulation models in a co-simulation environment (FMI for Co-Simulation). Co-simulation is a rather general approach to the simulation of coupled technical systems and coupled physical phenomena in engineering with focus on instationary (time-dependent) problems. FMI for Co-Simulation is designed both for the coupling of simulation tools (simulator coupling, tool coupling), and coupling with subsystem models, which have been exported by their simulators together with its solvers as runnable code.”

This type is shown in Figure 5-48.

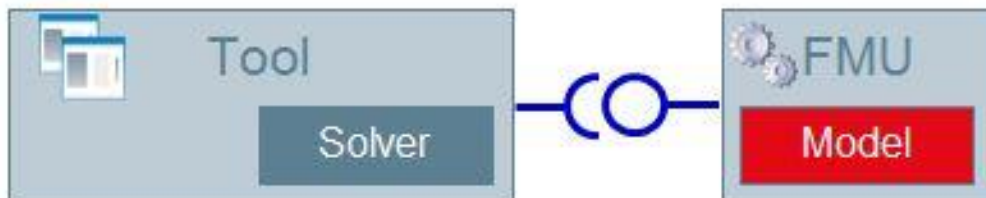


Figure 5-48 MODELISAR FMI for Model Exchange (Blochwitz & Otter, 2011)

- MODELISAR FMI for co-simulation discussed in the MODELISAR Consortium (September 30, 2010):

“[...] defines the “Functional Mock-up Interface for Model Exchange”. The intention is that a simulation authoring environment can generate C-Code of a dynamic system model that can be utilised by other simulation authoring environments. Models are described by differential, algebraic and discrete equations with time-, state- and step-events. The models to be treated by this interface can be large for usage in offline or online simulation or can be used in embedded control systems on micro-processors. It is possible to utilise several instances of a model and to connect models hierarchically together. A model is independent of the target simulator because it does not use a simulator specific header file as in other approaches.”

This type is shown in Figure 5-49.

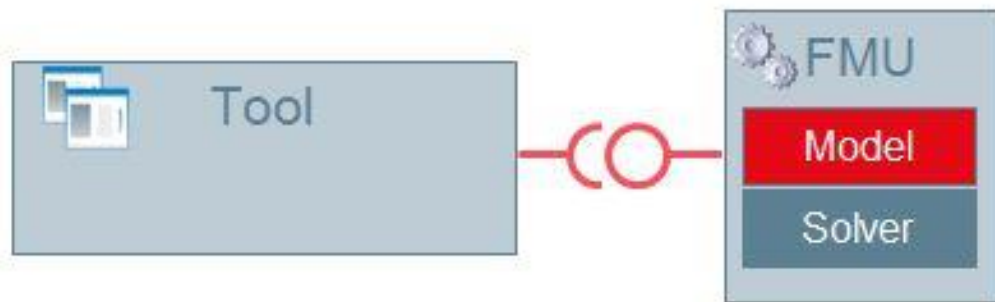


Figure 5-49 MODELISAR FMI for Co-Simulation (Blochwitz & Otter, 2011)

Additional discussions on the usage of MODELISAR FMI and FMU were completed by the MODELISAR Consortium (January 26, 2010):

“The FMI (Functional Mock-up Interface) defines an interface to be implemented by an executable called FMU (Functional Mock-up Unit). The FMI functions are used (called) by a simulator to create one or more instances of the FMU, called models, and to run these models, typically together with other models. An FMU may either be self-integrating (co-simulation) or require the simulator to perform numerical integration.”

Both types of MODELISAR interfaces can be used for the system-simulation generation of the electrical front door window lifter system. What is of interest to the automotive company is the coupling of the three simulation disciplines, rigid body simulation, flexible body simulation and logical simulation, via MODELISAR interfaces. In the earlier discussions, three main sub-simulation models, used for the electrical front door window lifter system example, were discussed. However, each of the sub-simulation models was solved by another solver. In the case of the logical simulation discipline, additional sub models emerged (see Section 5.5.3 I). It has to be mentioned that this system simulation description is used for a specific stage in the development process. At other stages in the development process, this structure can be different. For system-simulation generation, a hierarchical interaction of MODELICA

simulation models can be achieved using MODELISAR FMI for Model Exchange. This interface forces a direct interaction between the Modelica c-source-codes and the Modelica simulation models.

In order to connect the flexible body simulation and rigid body simulation, the MODELISAR FMI for Model Exchange is not useable. The flexible body and the rigid body simulations have to use their specific solver and the solver has to be integrated into the MODELISAR interface. So, the MODELISAR FMI for co-simulation has to be used. Sub simulations within MODELISAR FMI are assembled and ordered in a hierarchical structure. This structure is shown in Figure 5-50.

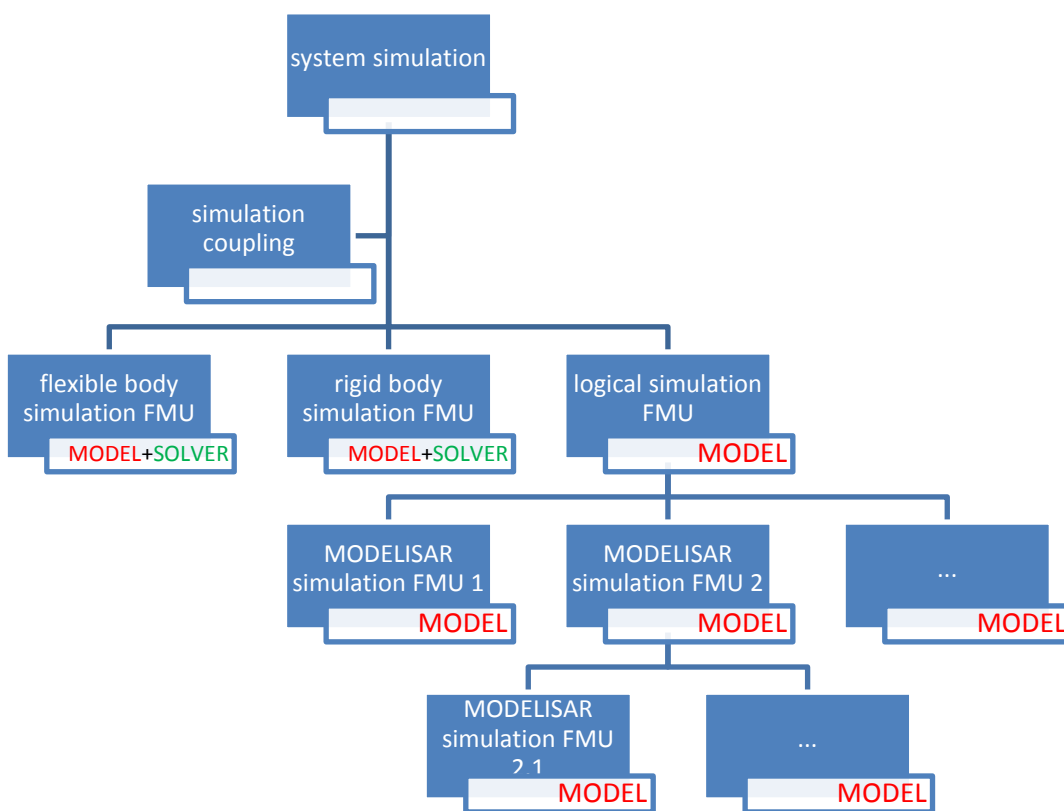


Figure 5-50 FMU-Structure of Simulation of Case 4

Data management should support this kind of system-simulation. This will be discussed in Section 5.5.4

#### 5.5.4 DATA MANAGEMENT OF CASE 4

This case study does not include an integration of the system-simulation example into the PLM framework TEAMCENTER. A system-simulation of the electrical front door window lifter system is produced, but this system-simulation is not viable due to missing functionality

between co-simulation interfaces in the simulation authoring tools. Data management of the system-simulation has no chance of improving this situation. So, the integration of the system-simulation example into the PLM framework TEAMCENTER could not be completed.

In addition to this, the changing system-simulation during the development process of the electrical front door window lifter system was not transposed. This means that changes of the system-simulation structure and simulation models during the development process were modelled. However, the new framework should support this kind of development process, that is, the “system-simulation-based system engineering development methodology”. Therefore, data management has to support two areas of functionality:

- 1) Data management of the changing system-simulation and sub-ordered simulation models:

The use of system-simulations during the development process causes source changes in system-simulations. Product data becomes more detailed at each progressive level in the development process. This product data is the source for the simulation models. In order to achieve an improvement in the result quality (to bring them closer together with the reality), these simulation models should be based on improved product data. Thus the simulation models can be revised and based on improved product data, or the simulation models can be regenerated based on this improved product data. Simulation models can be single simulation models as well as system-simulations. This was discussed in Section 5.5.2 III.

- 2) Data management of the system-simulation variants due to multiple simulation models describing the same subsystem:

Dependent on the requested system-simulation results, simplified sub-simulation models can improve system-simulation performance. Sometimes, these simplified sub-simulation models are generated at early development stages. They can be overworked. Such simplified simulation models can be more economical for system-simulation. This causes the multiple variants of simulation models that describe the same sub-system to appear in a system-simulation.

The system-simulation requires the support of these different variants and versions of the subsystem-simulation models in the system-simulation structure. It follows that each, system-simulation is given its own unique version and has a positive influence on the management of the simulation results. The simulation results have to be traceable to the simulation models responsible for the results and to the system-simulation.

The new framework for the support and management of multi-disciplinary simulation data embedded in a PLM environment can be used to solve this problem. In Section 5.5.5, Case 4 will be empirically integrated into the new framework.



### 5.5.5 EMPIRICAL VALIDATION OF THE NEW FRAMEWORK AND CASE 4

The new framework for the support and management of multi-disciplinary simulation data embedded in a PLM environment can be used to enhance the actual possibility of multi-disciplinary and inter-disciplinary simulation data management. The files, data and information should be uniquely dependent and traceable.

An empirical start of the product development process can be the generation of functions and a functional structure (light blue in Figure 5-51). This provides a base for behaviour simulation models realizable in tools such as Modelica or Matlab/Simulink (light green in Figure 5-51). During the development process, these functions and functional structures can be improved and revised (dark blue in Figure 5-51). The improved functions and functional structures provide a base to improve behaviour simulation models (dark green in Figure 5-51). Further in the development process, the product describing data is produced. Such data can be in the form of CAD models (light orange in Figure 5-52). This data is detailed step-by-step in the development process (dark orange in Figure 5-52). Parallel to the modelling and designing, simulation models based on the product data, such as CAD models, are generated (yellow in Figure 5-52).

The paragraph above demonstrates that multiple simulation data is generated during the development process. Not all are useable for system-simulations. In order to make those useable simulation models identifiable for system-simulation, they are linked by a system-simulation source to their source base. A system-simulation target relationship helps to identify and trace the simulation models representing a system or product (blue connections in Figure 5-53). In order to depict the system-simulation source and target relationship, the functions, function structure, products and product structures are positioned to the right and the simulations, independent from the source and discipline, are positioned to the left in Figure 5-53.

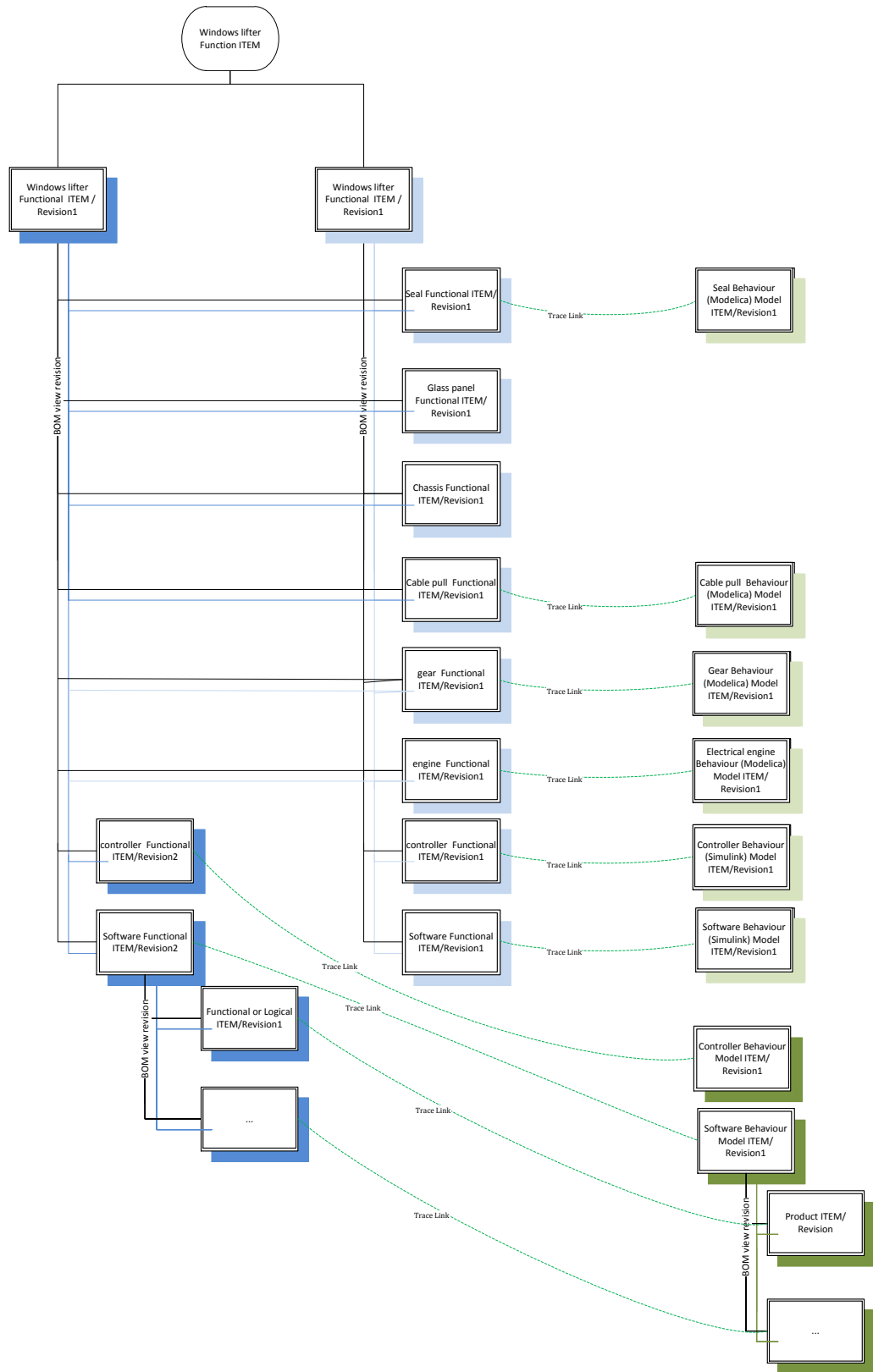


Figure 5-51 Functions and Behaviour Models during the Development Process

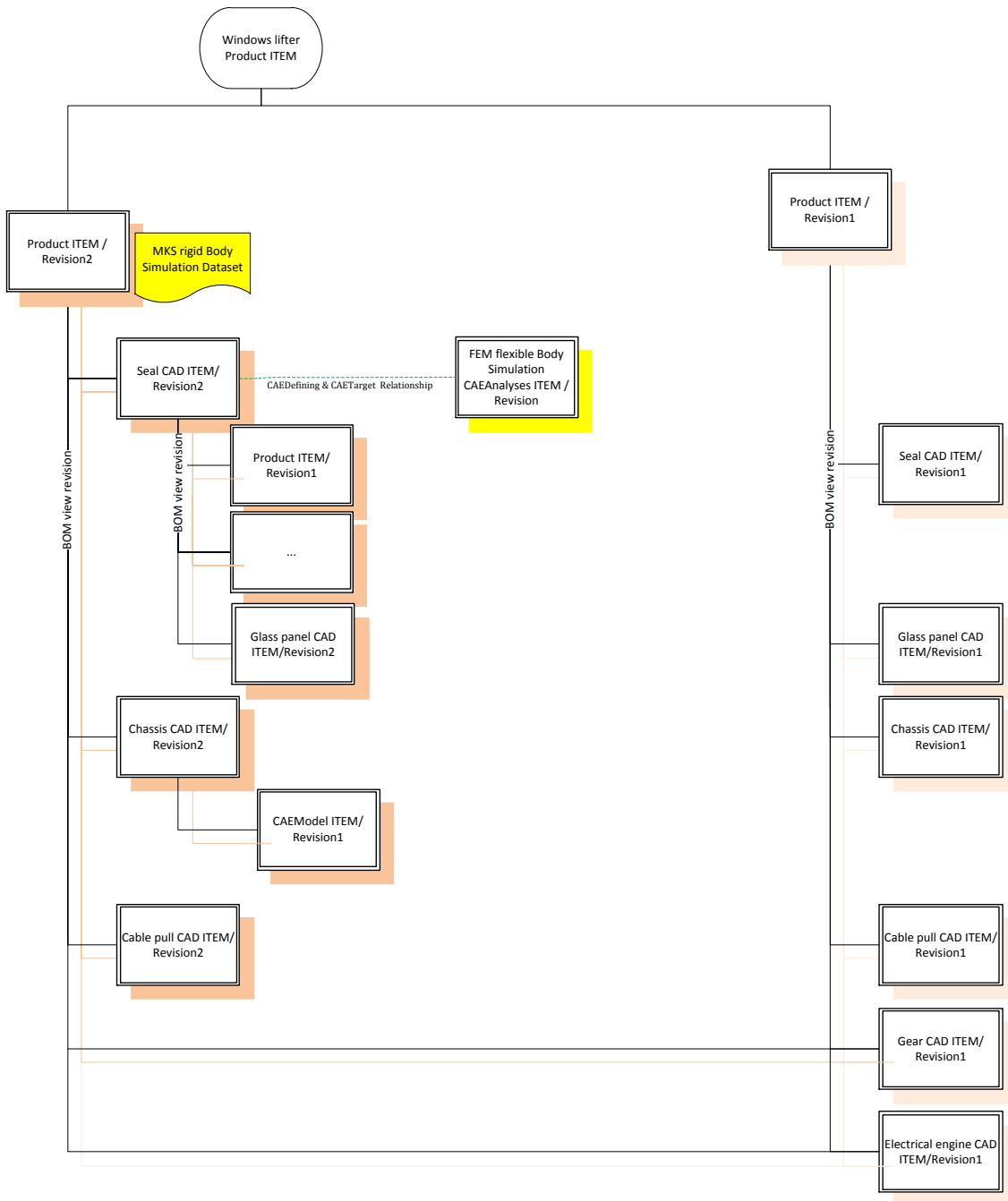


Figure 5-52 Product and Simulation Data during the Development Process

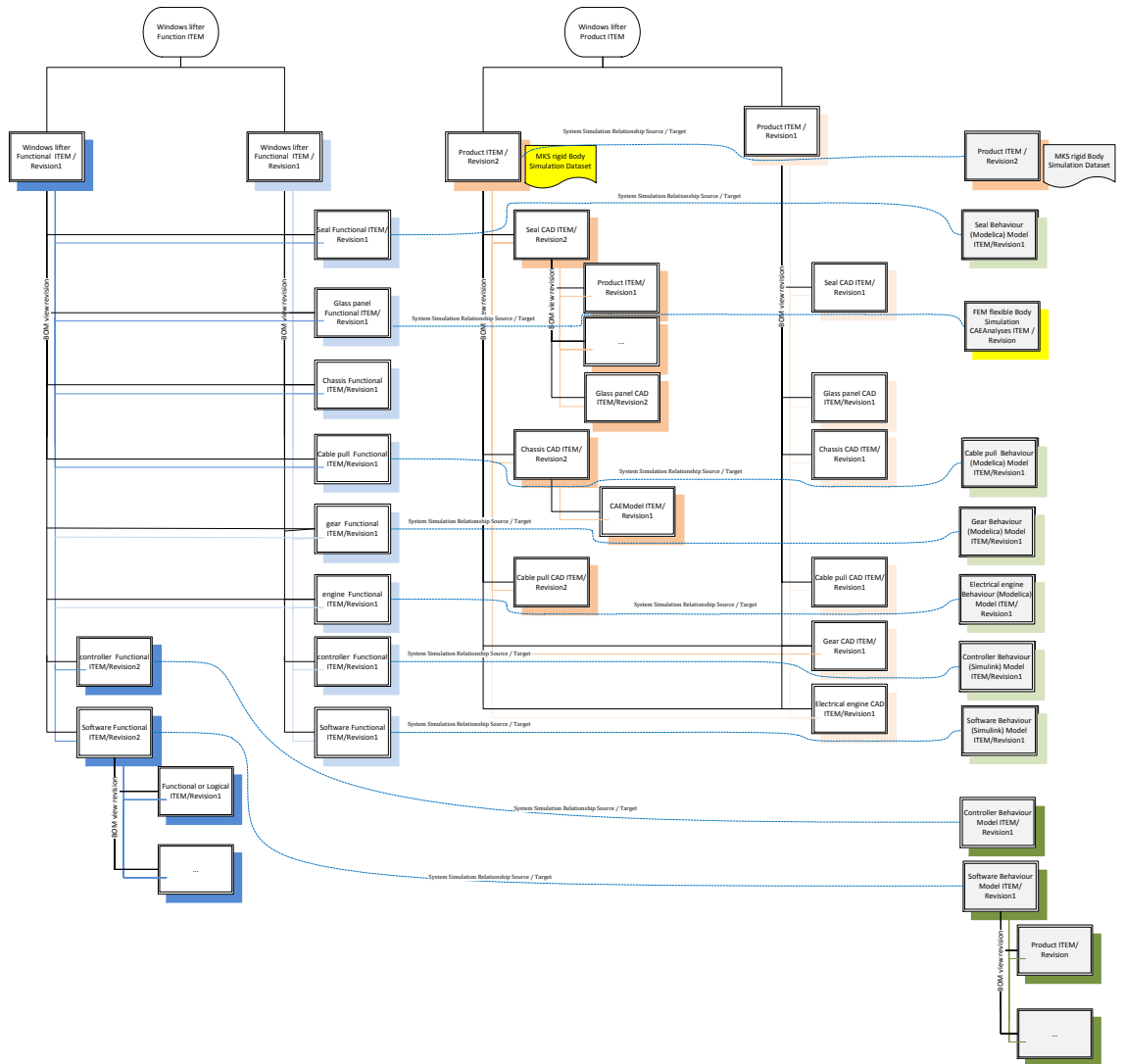


Figure 5-53 Simulation Model Traceability of Represented and Sourced System

The simulation models, identifiable so as to represent a product or function in a system-simulation approach, are used in sub-ordinated way in the system-simulation structure. The system-simulation structures sub order the different simulation models by using BOM technology. The simulation discipline, simulation authoring tool or item type managing the simulation model are not important. The available simulation models, useable in a system simulation approach, are more EBS discipline-oriented and managed as behaviour models at an early development stage (see System-Simulation item revision 1 in Figure 5-54). With an ongoing product development process, other simulation models can replace previously generated simulation models. These simulation models can be more geometrically based such as rigid body or flexible body simulation disciplines. So, the system-simulation structure has to be revised to create a changed version including other sub-ordered, more recent, simulation models (see System Simulation item revision 2 in Figure 5-54). The system-simulation structure may be required in different versions. One version might include more detailed, sub-ordered simulations whereas other versions might use another simulation or a curve (see Variant A including the electrical engine as behaviour simulation and Variant B excluding the electrical engine simulation; pictured in Figure 5-54). The revision and version changes of the system-simulation structure create simulation results. The simulation results have to be traceable back to the simulation models responsible for the results and to the system simulation. With the generation of extracted system simulation result items, the management of the results are dependent on the revision or version of the system-simulation structure achieved (see Figure 5-54). These system-simulation results can be organised to represent a system-simulation result structure equivalent to the system-simulation structure. Traceability is achieved via a CAEResult Relationship pointing from the system simulation result to the source or target. This kind of simulation result management supports the generation of system-simulation results for each version and revision of the system-simulation structure. Multiple system-simulation results can be managed for the same version or revision, which may be needed for system-simulation parameter studies.

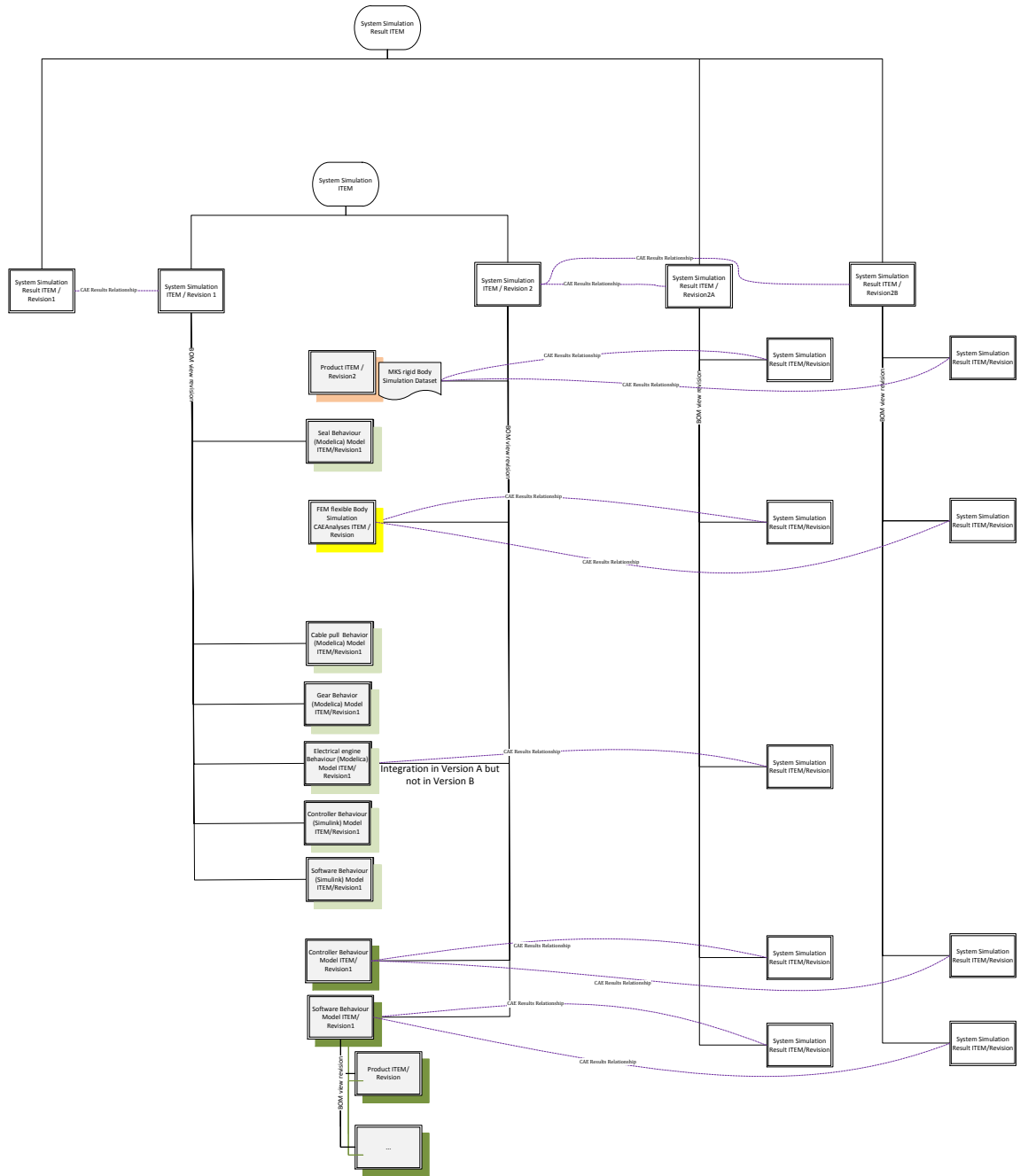


Figure 5-54 System-Simulation and Result Management

As discussed above, the findings from the four case studies can be used to generate, validate and verify the new framework. The new framework will be discussed in the Chapter 6.

## 6 A NEW FRAMEWORK FOR IMPROVING THE SUPPORT AND MANAGEMENT OF MULTI-DISCIPLINARY SIMULATION DATA IN A PLM ENVIRONMENT

A new framework can be used to improve the support and management of multi- and inter-disciplinary simulation data. The first and foremost improvement should be the integration of the new framework into a PLM system. Other improvements can be achieved by optimizing technologies and architectural designs. In Section 6.6, these improvements will be discussed.

### 6.1 INSTITUTION OF A SYSTEM-SIMULATION STRUCTURE

From a traditional, administrative point of view, a multi-disciplinary simulation is organised as a process (van Beek, Rooda, Engell & Zaytoon, 2000). This kind of view is popular due to the integration of multiple models, singularly generated by specialised departments and experts, into a simulation process. In contrast, the behaviour system-simulation methodology uses a simulation structure to manage sub-ordered simulation models. In the case of the behaviour methodology, the top and sub-ordered simulation models are generally modelled using the same simulation authoring tool. The simulation authoring tool used independently of simulation structure can improve the approach. In Section 6.1.1, the case examples will be analysed from this perspective.

#### 6.1.1 ANALYSIS OF THE CASES

In Cases 3 and 4 discussed in Chapter 5, the simulation models are generated independently, subsystem-oriented and exchangeable. The sources of the system-simulation models are CAE simulation models (managed in SDPM or PDM) as well as behaviour simulation models (managed in behaviour data management). Afterwards, the subsystem-simulation models are coupled or merged. This coupling and merging generates the system-simulation. The system-simulation is based on a system description including subsystems. Each of the coupled or merged simulation models simulates a specific component or subsystem of the system. This subsystem or component can be based on specific functionalities or specific assemblies of the system. The summary of the coupled or merged simulation models represents a subsystem or system. An example of a system simulation that is based on system architecture is illustrated in Figure 5-27. The figure shows which system component is represented by the simulation model and which merging hierarchy is used to generate the system-simulation model.



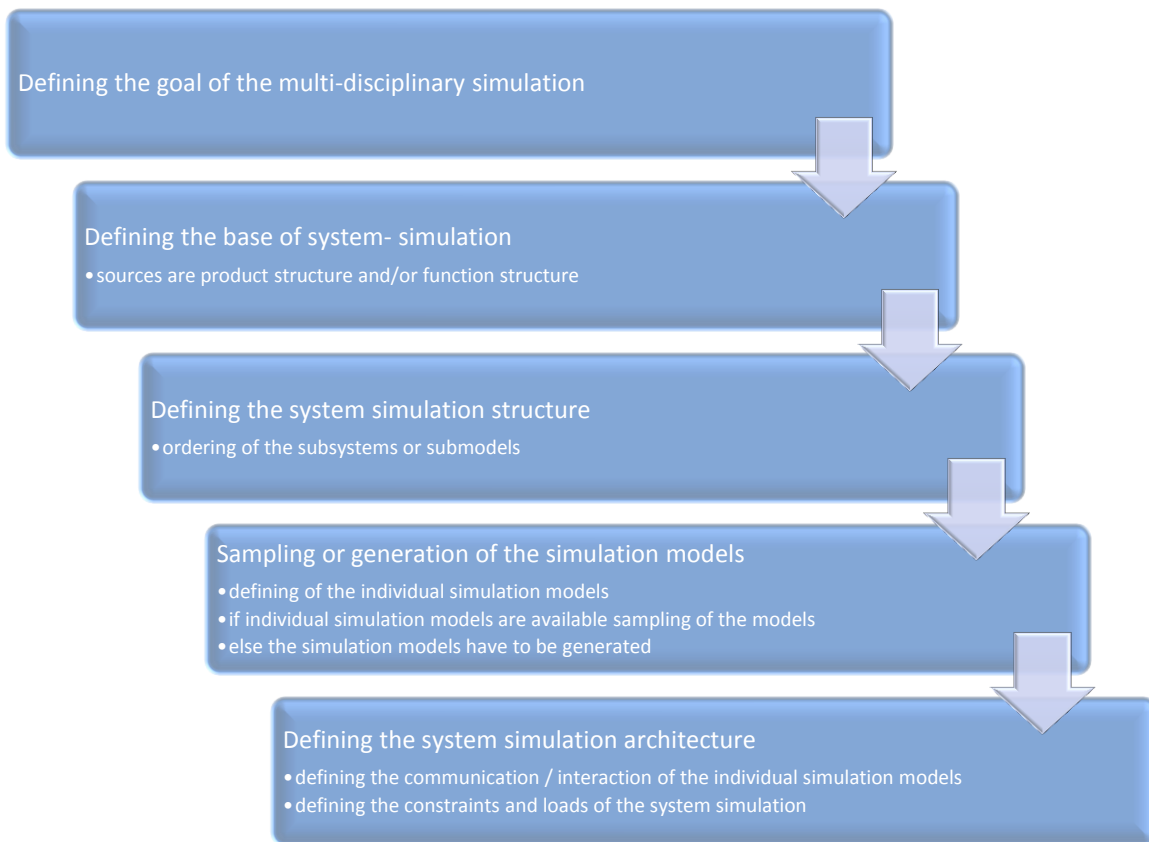
Another example is represented in Section 5.5.5. So, the simulation models can be generated individually and then interact with each other afterwards in a system-simulation structure.

A multi-disciplinary simulation consists of systems, products or processes. The individual simulation models can be built based on a hierarchy of systems, products or process. So, the multi-disciplinary simulation is usually a system-simulation. The sources for the simulation models are descriptions of the products, functions and processes. In all case studies, the dependencies on the product description are given. In Cases 3 and 4, the dependency on the functional descriptions is also used to generate the simulation models. Especially in the early development phase, the functional structure is used to generate the system-simulation structure such as in Case 4.

Based on product or function description, the individual simulation models can be (pre-) defined. These defined individual simulation models can be arranged into a system-simulation structure. This kind of arrangement has been discussed in all case studies. Here, the product and/or function structure is congruent to the system-simulation structure. In addition, any existing simulation models can be re-used and ordered into the system simulation structure. This requires the identification of simulation models based on the source data such as function or product items. The identified simulation models have to be checked from the perspective of reusability to see if the requirements for the new multi-disciplinary simulation are fulfilled. Case 4 demonstrates that behaviour models are re-used for later system-simulations. If any simulation models are missed, new simulation models have to be generated or existing ones have to be modified. Changes to existing simulation models have to be based on the sub-product or sub-function that should be represented by the new simulation model. These kinds of changes were carried out in Cases 3 and 4 where the NX Motion simulation model was generated as an independent simulation to investigate the mechanical property of the product. The existing simulation of the mechanics was changed in some areas to make it work as part of a system-simulation. So, the system-simulation structure can be used to organise the administrative work in more efficient way.

Dependent on the integrating simulation middleware (such as was necessary in case example 4) or simulation authoring tool (such as NX Motion in case example 3), the integrating simulation model data has to be managed. Such a simulation authoring tool or simulation middleware acts as a simulation integrator coupling different simulation models. Due to the integrator functionality, these kinds of simulation models represent the top level of the system-simulation structure. So, the system-simulation structure can help to outline the administrative simulation work more easily.

Case examples 1 and 2 are different at this stage. They do not fulfil the necessity of linking individual simulation models to be run as co-simulations. These two case examples describe a system-simulation as a serial process. In contrast to the first two case examples, case examples 3 and 4 describe system-simulation examples in a parallel process. In the following, the system-simulation of a system-oriented multi-disciplinary simulation process is discussed to achieve a more unique understanding.



**Figure 6-1 System-Oriented Simulation Process**

The generation of system-simulations assembles multiple individual simulation models to a system-simulation structure. The system-simulation structure describes a system, not a process. The system-simulation structure can be based on the system description. Afterwards, the required and useable individual simulation models can be identified or generated. These simulation models have to be integrated into a co-simulation. The run of the co-simulation requires multiple solvers that communicate with each other. Each individual simulation model is run with its own solver properties. A simulation middleware or simulation solvers, that provide interfaces to other simulation solvers, take the data provided by the individual simulation solver and provide data needed by the individual simulation solver. Each simulation solver produces simulation results or protocols. The review of these simulation results or protocols is completed in the post-process. However, the review of results is dependent on the simulation model's authoring tools. Some authoring tools provide general result formats, others, specific formats. The understanding of the results is dependent on the expertise of the reviewer. Accordingly, multiple experts and post-processors are required (Zaeh & Baudisch, 2003). Also, the simulation integrator (such as simulation middleware) can produce result and protocol files. Usually, this data is focused on

parameters and stored in a more general way. The reviewer does not need special expertise of specific simulation authoring tools to be post-processed. The previously described system-simulation process is shown in Figure 6-1.

### 6.1.2 *THE SYSTEM-SIMULATION STRUCTURE APPROACH*

The new framework should support the interaction of system-simulation structure, simulation model sources and a simulation integrator such as described in Section 6.1.1. The three cores - system-simulation structure, system model sources and simulation integrator, interact with each other. The system model sources can be understood as databases storing simulation models in their files and metadata. The system-simulation structure should be understood as hierarchical descriptions of the required simulation models and a database for the system-simulation integrator data. The simulation integrator is a simulation middleware or a simulation authoring tool with the functionality to integrate different simulation tools or data to a common or interacting co-simulation. These are shown in Figure 6-2.

The new framework makes it possible to search for simulation models in the simulation model sources. After checking the usability of the models for re-use, these models should be linked as sources to the system-simulation structure. It is possible that not all required simulation models in the system-simulation structure are identifiable. When simulation models are missed, a metadata set in the simulation model sources should be added and linked to the system-simulation structure. The added metadata set is an empty container and will be filled later with the required simulation model data and files. The generation of the individual simulation model and other required files are carried out in/by the specific simulation authoring tool by a responsible person with the specific expertise (Zaeh & Baudisch, 2003).

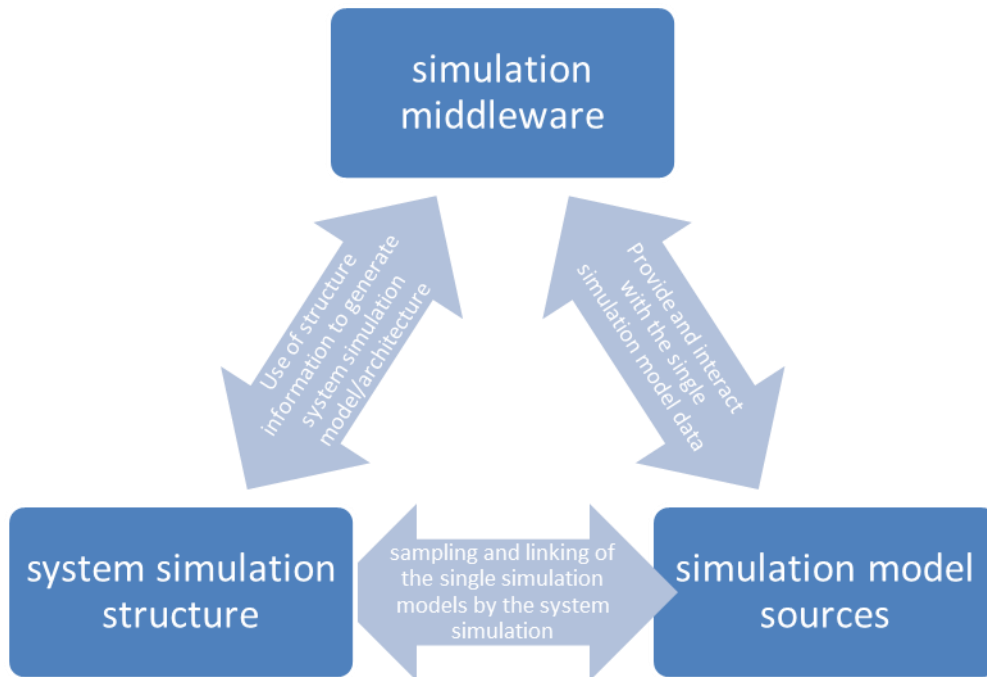


Figure 6-2 Multi-Disciplinary Simulation Data Management Cores

Structural handling of the data is a technique for structuring the previously described data management. With this handling of the models, the new framework can be used:

- to reduce data repetition by integrating all required data into one database. The database has to combine PDM, SDPM and behaviour model management with system-simulation data management. Additionally, the product and functional descriptions have to be integrated into this approach. All this can be addressed by a PLM system.
- to support the sampling of simulation models by linking metadata information. The simulation models are managed by different data holding approaches. Such data holding approaches can be SDPM, PDM and behaviour model management. Additionally, all these data holding approaches can also be used to manage metadata information. In the case of searching and sampling, the metadata can be re-used. This requires one common and/or multiple interacting databases.
- to improve the administration of the simulation models and sources in the context of multi-disciplinary simulation. Such an administrative improvement is, for example, traceability of the system-simulation structures to their sources as well as to the interacting simulation models. The traceability can be achieved by linking dependent metadata information. The link can be followed to check, for example, the veracity of the source.

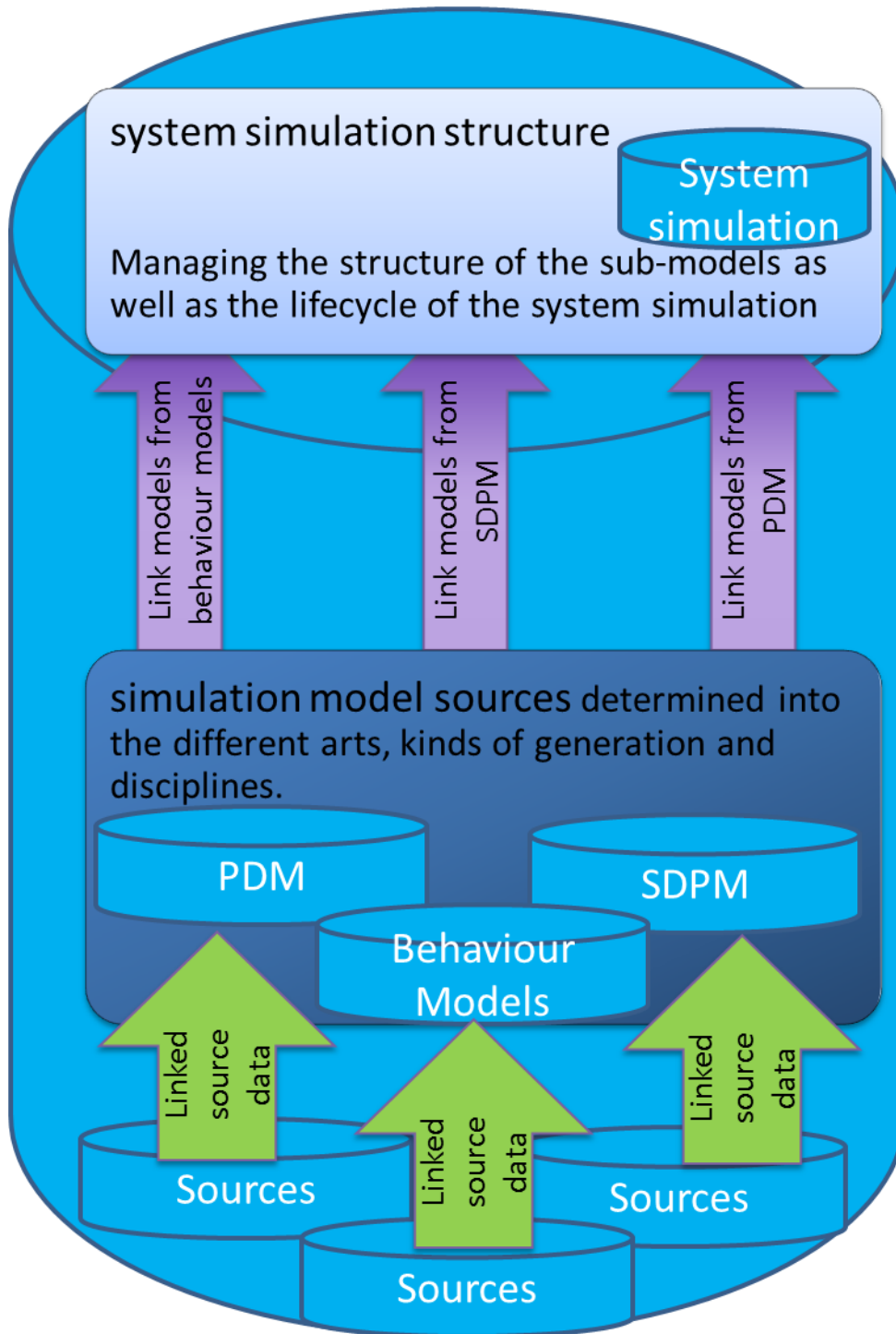


Figure 6-3 Improved Approach of System-Simulation Source Dependencies

The previous analysis (especially in Cases 3 and 4) as well as the process discussions results in a requirement to adapt system-simulation structures. System-simulation structures can interact with other system-simulation structures (sub-ordered) and simulation models. Sources for the simulation models can be SDPM or PDM and behaviour simulation model

management approaches. The sampled simulation models are linked to the system-simulation structure. So, the data does not need to be duplicated and is uniquely traceable. The simulation models should relate to their own sources such as product or function descriptions. So, there is also a link to their sources. Such an approach is part of common development strategies and most effective in combination with PLM approaches. PLM systems can provide multiple data (such as PDM, SDPM and behaviour model management, data management of product and function descriptions). This approach is shown in Figure 6-3.

This framework is a new and improved approach, when compared to current system-simulation managing tools such as the IMAGINE.Lab SysDM application from the LMS Company. System-simulation managing tools are not PLM-implemented tools and focus on the support of behaviour or one-dimensional-simulation models. A unique traceability to simulation model sources cannot be achieved due to the PLM native approach. So, there are no interactions between system-simulation data managing approaches and other data holding approaches, and data has to be copied and duplicated from other sources. The focus of such tools on behaviour or one-dimensional-simulation models results in a lack of support from other simulation disciplines. This type of approach is shown in Figure 6-4.

With the new approach, a copying and duplication of simulation models and other sources would no longer be required. The system-simulation structure makes the sub-ordered simulation models traceable by linking their metadata. The new framework keeps the simulation models traceable to their sources.

For example, an empirical system-simulation structure consists of an MBS model and an EBS model. Both are sub-ordered within the system-simulation structure. The data of the MBS model is managed by PDM. The CAD model sources are also managed by PDM. The EBS model is managed by behaviour model management and dependent on a specific function. The new approach achieves the traceability from the system-simulation structure to the individual simulation sources. This example is shown in Figure 6-5. Similar examples are given in Case 3 (see Section 5.4.5) and Case 4 (see Section 5.5.5).

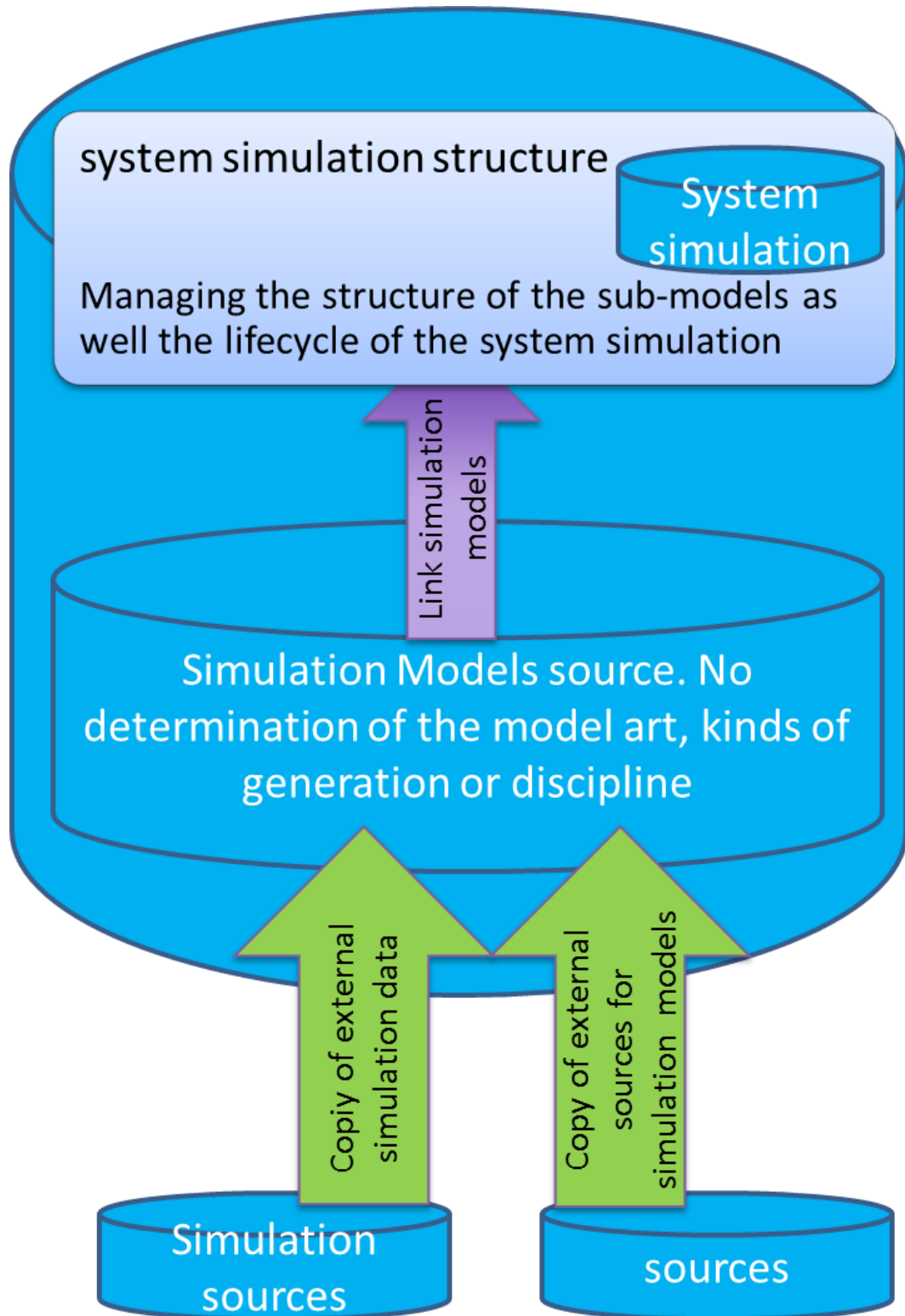


Figure 6-4 Existing Approach of System-Simulation Source Dependencies



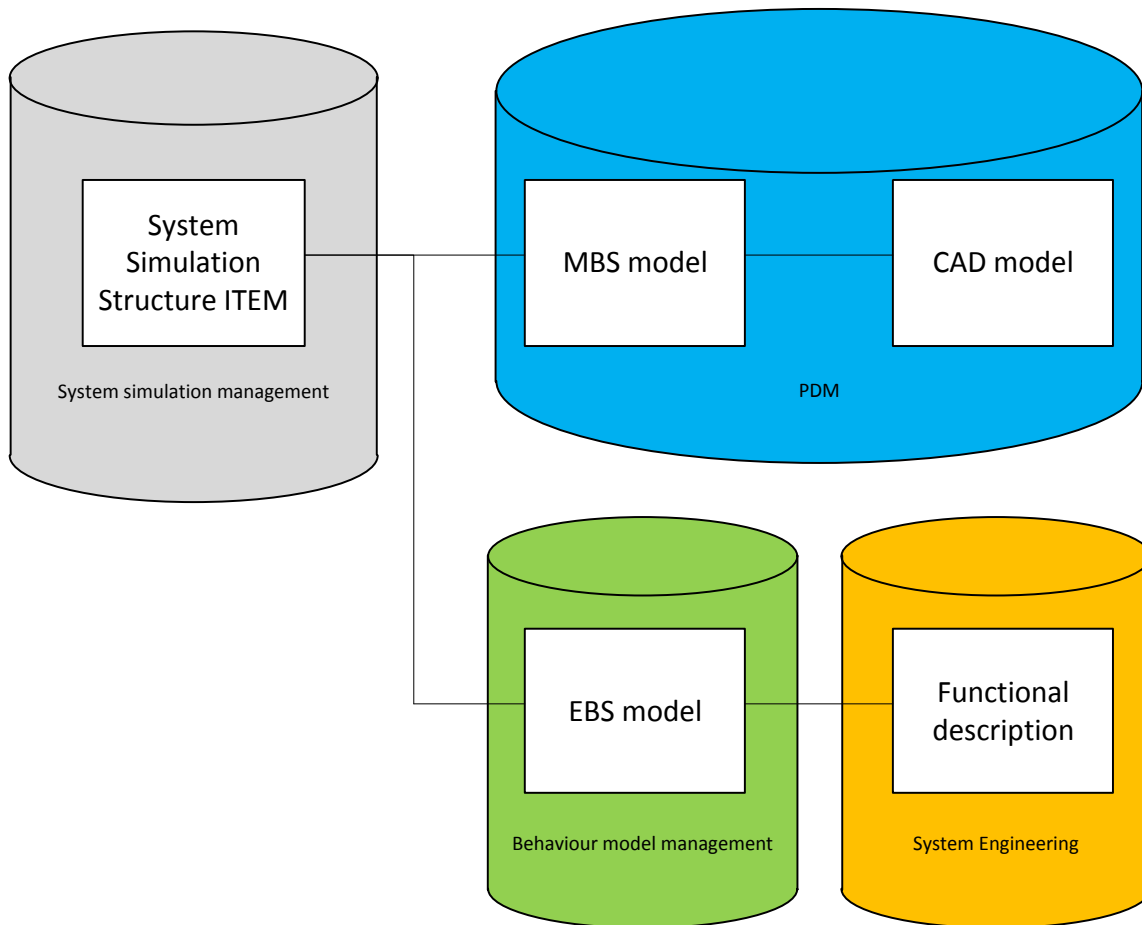


Figure 6-5 Traceability-Example of the New Approach

The system-simulation structure has to provide the support of the system lifecycle during the development process, i.e. changes to the system-simulation during the development process have to be managed. So, a revision of the system-simulation structure is required. TEAMCENTER provides the revision with standard functionalities. Section 6.1.3 will discuss the new approach in more detail.

### 6.1.3 TECHNICAL DETAILING OF THE APPROACH

The system-simulation structure can be similar to a product structure management within TEAMCENTER. TEAMCENTER is a PLM system and useable as the integration platform for the new framework. In TEAMCENTER, the base for the data ordering and storing is a single object called Item.

“Items are the fundamental objects used to manage information in TEAMCENTER. They represent parts and other objects that you want to manage within a lifecycle. Items generally represent data that is configuration controlled by revisions. Items collect a variety of different types of business data, for example, CAD design files for parts, document files such as specifications, reports, and forms for metadata. An item can be thought of as a package that contains all data related to that item. Each item has at least one item revision and a label containing two pieces of information:

[...]

Overview of product structure

- Item ID  
A unique identifier for the item. No two items can have the same item ID. An item ID may be the part number or the document number of the object it represents.
- Item name  
A short description that is usually a logical name such as Bolt, Bracket, or the title of a document.

The term Item generically describes all types of items that exist in TEAMCENTER. To effectively manage many types of item, you should create specific types of item appropriate to your business.

You should also distinguish between the item and its associated item revisions, as follows:

- Item  
An item commonly represents manufactured product such as parts, assemblies, end items, and tools. It is an abstract container that holds item revisions and general documents that apply to the product, rather than to a particular revision. You cannot build or test an item.
- Item revision  
An item revision represents a physical entity and is a unique, specific revision of a previously created item. It may have associated CAD models, drawings or specifications that are applicable only to this revision. You can release an item revision with a workflow or through change management. This action applies a Released status to the item revision, preventing further edits and allowing TEAMCENTER to maintain product history (Siemens Product Lifecycle Management Software Inc., 2012).”

This is useful in finding the right object to describe a system-simulation structure. The above extract mentions that items and item revisions have to be specific. In the case of system-simulation, a new item and item revision is required to represent the system-simulation structure. In this thesis, this will be called item ‘Sys-Sim item’ and the item revision ‘Sys-Sim item revision’.

The product structure begins with a top item Revision such as a CAD item revision. A structure is created by sub ordering additional item revisions. The sub-ordered item revisions can be a structure on their own including further sub-ordered item revisions. The depth of the structure is limitless. This means of building structures is also required for the system-simulation structure. The means of structuring is achieved by a BOM View Revision.

“[...] you create a product structure, sometimes loosely called a bill of materials (BOM).

[...] In their designs, engineers group parts together in assemblies to allow re-use of the same assemblies elsewhere in the product or in other products. An assembly can contain components that are piece parts or other assemblies. In this way, you can model a complete product structure as a hierarchy of single-level assemblies. From the TEAMCENTER perspective, piece parts and assemblies are both represented by items, and each item has at least one revision (Siemens Product Lifecycle Management Software Inc., 2012)."

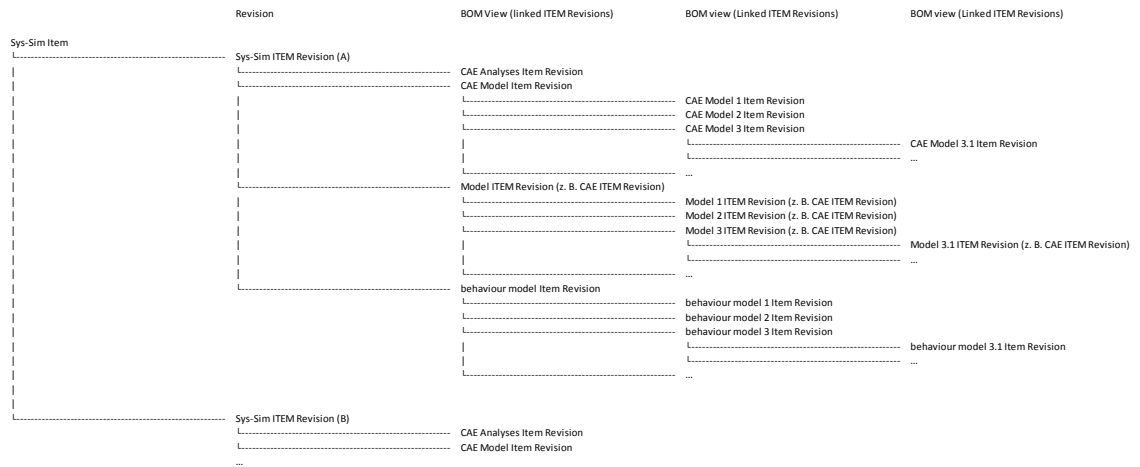
One requirement of the system-simulation structure is the reusability of individual simulation models. This is similar to the re-use of components and parts in the product structure. The BOM view technology of TEAMCENTER can also meet this requirement.

"You can create assemblies that are precise and reference a specific revision of each component; you can also create dynamic assemblies in which current revision rule determines the configuration of the assembly. Dynamic assemblies are sometimes referred to as imprecise assemblies. The hierarchical structure relationship between the immediate parent assembly and its child component item or item revision in a precise assembly is represented by an occurrence (sometimes called a relative occurrence) revision (Siemens Product Lifecycle Management Software Inc., 2012)."

This perspective was not included in the case studies, but it is realistic in the case of using system-simulation structures. In the case of freezing a system-simulation structure, which means that a product goes into production and the data remain unchanged, the system-simulation structure has to be reloaded precisely (dependent upon the frozen state). However, the product will change for example, the design, parameters, materials or supplier. Such design improvements could be caused by product optimization or supplier changes. This will cause the product description to change. Therefore, the system-simulation structure needs to make it possible to be loaded non-specifically.

"When you add a component to an assembly, you are creating an occurrence of that item or item revision in the assembly, which is stored on the BOM view revision. This occurrence is displayed as a BOM line. A BOM view revision is a single-level structure that contains occurrences of its immediate children. A multilevel structure is built up from many single line BOM view revisions (Siemens Product Lifecycle Management Software Inc., 2012)."

The ordering possibility of a BOM view technology is monitored as a neural example in Figure 6-6.



**Figure 6-6 Ordering of the System-Simulation Structure**

The technology of the BOM view is enormously useful for the system-simulation structure. The BOM view is independent from the used items and item revision types. So, different items and item revisions can be used for the Sys Sim item and Sys Sim item revision. However, the Sys Sim item revision sub orders different item revision types. Different item revision types have to be sampled by the BOM view of system-simulation structures because the different simulation models are managed under different item types. The different item revision types are listed in Table 6-1 Item Types used in the System-Simulation Structure.

**Table 6-1 Item Types used in the System-Simulation Structure**

Source	ITEM Type
Simulation or simulation structure managed in SDPM	CAEAnalysis ITEM Revision or CAEModel ITEM Revision
Simulation models managed in PDM	ITEM Revision (for example the CAD ITEM revision mentioned in the case studies)
Behaviour model as simulation model	Behaviour model ITEM Revision

The SDPM technology of TEAMCENTER manages simulation models via two different types of item. These may be a CAE Analysis item revision and CAE Model item revision. In the case of a CAE Analysis item revision, no direct structuring is available. It is possible to follow dependencies of the CAE Analysis item revision to get more details, but there is no BOM view available. In the case of a CAE Model item revision, a BOM view can be available. However, the BOM view is not a requisite for the CAE Model item and CAE Model item revision.

This is similar to simulation models in the PDM environment. In the PDM environment simulation models can be stored as datasets under a CAD item revision or other item revisions. An item revision can include a BOM view as discussed above. However, the

simulation model managed under an item revision is usually independent from the BOM view of the item revision.

The behaviour simulation models are managed as behaviour model item revisions. They can be sub-ordered by a BOM view. This sub-ordering is used in Cases 3 and 4 for the software and controller unit. In this case, the BOM view of the behaviour model item revision can be used in the system simulation structure to get more details about the simulation model.

A representation of the system-simulation structure is achievable with:

- the technology of items and item revisions to generate the data managing objects and
- the BOM view technology to manage the sub-ordered simulation models.

However, the system simulation structure can change during the lifecycle. Two main reasons are:

- the first one is the historical change of system-simulation structures. Such historical changes can be the changes during development processes.
- the second one is the representation of product variants.

The historical change is caused by the on-going development process of the products. Changes to a product can cause changes to the simulation model representing the product. This means that a simulation of the product or sub-product in multiple stages of the development process might be required. An example of this is Case 4. In this case, simulation models of an early development process as well as simulation models of a late development process have to be integrated into data management. In order to manage the change of the data, the revision technology is used by TEAMCENTER (see Section 4.1). If a simulation is frozen, no changes are allowed. However, the simulation source (e. g. CAD model) data does change and leads to the new item revision. The analyst has to decide if the frozen simulation is valid or if the simulation has to be revised. The editing of the simulation model can be done on the revised level whereby the original will be saved. Therefore, the data management of simulation models is also manageable with the revision technology (see Section 4.1). The system-simulation model or the system-simulation structure can be edited using revision technology.

In the publications of Zehetner, Wenpu Lu, Watzenig & Bernasch (2012) and of LMS International (2011), the integration of version management and variant management was deemed an important function. The representation of variants only arose in a reduced dimension in Case 4. However, IMAGINE.Lab SystemSynthesis (LMS International, 2011) also provides the possibility to build variants of the system-simulation structure. These variants are used to exchange a subsystem by using another equivalent subsystem. Variants can be used, for example, to build a system-simulation of the thermal management of a vehicle where a subsystem (for example, the combustion engine) is exchangeable (for example, different power intensive combustion engines). TEAMCENTER also provides the

possibility of variant management based on a structure. The re-use of this technology for a system-simulation structure has the functionality to build multiple variants based on one system-simulation structure. Additionally, the variant management of TEAMCENTER does not require multiple versions of the structure. Using a 150% structure and configuring the structure achieves the data management of a configuration. Filtering the structure based on the configurations results in multiple variants of the structure.

“You can define sophisticated revision rules that allow you to configure the structure in different ways. This allows you to create a single structure and re-use it many times, for example, for different versions of the product. Revision configuration depends on the release status of an item revision, and its related effective date, effective unit number, or the release date. It allows you to reproduce a configuration that was effective at a certain date in the past or recreate the configuration of a specific unit revision (Siemens Product Lifecycle Management Software Inc., 2012).”

The revision aspect in the variant technology of TEAMCENTER is well established and available in a wide range of TEAMCENTER use cases. This technology meets the necessary requirements of a flexible system-simulation structure. However, the existing technology of versioning and variant building is not the focus of this research. Additionally, the case studies of this research are not focused on improving this technology. In order to be focused on the main issues of this project, this technology will not be discussed in detail.

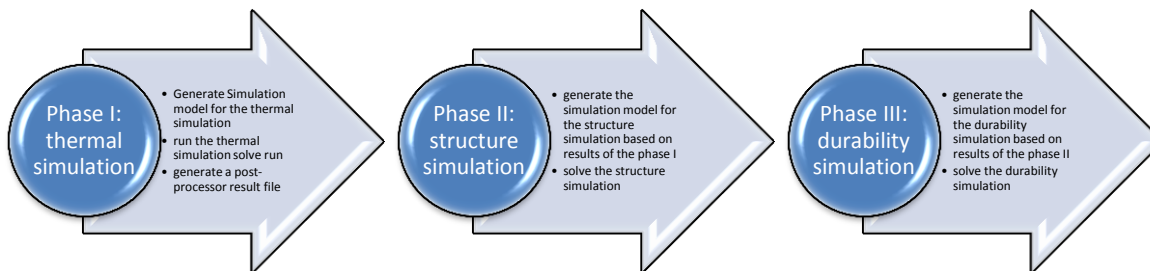
In Section 6.2, the difference between system-simulation structures and process-oriented simulation processes, and how system-simulation structures can support the process-oriented simulation process will be analysed.

## 6.2 COMBINATION OF SYSTEM-SIMULATION STRUCTURE WITH PROCESS-ORIENTED SIMULATION PROCESSES

As discussed above, from a traditional and popular administrative view, a multi-disciplinary simulation is organised like a process. Usually, the specialised departments and experts are focused on their specific problems and not on the system (Tian, Yan, Parkin & Jackson, 2008). This causes them to have a blurred system-view but a focused view on their simulation models and the interaction to other simulation models. They view the linking of individual models into a multiple-disciplinary model as a process. So, the traditional understanding of multi-disciplinary-simulations is more a simulation process than a system view. Nevertheless, the process-view can be suitable to system-simulation. The simulations responsible for multi-physical CAE-simulations often do not see the system-simulation from a system view. Section 6.2.1 will analyse the Cases from both system-simulation and simulation process points of view.

### 6.2.1 ANALYSIS OF THE CASES

Cases 1 (See Section 5.2) and 2 (see Section 5.3) can be organised as simulation processes. Figure 5-12 shows a serial simulation process from Phase 1 to Phase 3 (also shown in Figure 6-7). The phases can also be subordinated in serial simulation processes.



**Figure 6-7 Serial Simulation Process of Case 1**

Similar to Case 1, Case 2 is also a serial process. However, Case 2 can be parallelised. This is shown in Figure 5-18. The parallelisation is possible because there are two independent branches at the beginning. This parallelisation of the process ends when the branches have to be merged.

Both examples create multiple simulation models during the process. These simulation models are built and solved in the chronological order of the process. It is possible that this chronological process is parallelised. In the case of parallelisation, there is no dependency to other branches in the process.

Both simulations of Cases 1 and 2 are barely generated based on a system or subsystem structure. This is in contrast to Cases 3 and 4 analysed in Section 6.1.1. The focus of Cases 1 and 2 is process-oriented. The process can be a product development process or a production process. All of the Cases are multi-disciplinary simulation processes. However, the difference between Cases 1 and 2 and Cases 3 and 4 is the process or the system orientation.

The system-oriented Case 4 also includes a montage-oriented simulation process. The simulation process is shown in Figure 5-47. The seals of the electrical front door window lifter system are deformed by the montage. The simulation model reflects the virtual seals as unloaded. So, a pre-load causing the pre-deformation of the seals is required. This simulation process has to be based on the system-simulation where the flexible body simulation of the seals is involved. In Case 4, a simulation process and a system-oriented simulation are required.

Simulation processes and system-oriented simulations have to be supported by the new framework. However, the focus of the new framework is not on the support of simulation processes. The focus will be on the data management of the multi-disciplinary simulation. Multi-disciplinary simulation in this case is rather the data management of the system-simulation and system-simulation models. The new framework should support the re-use of



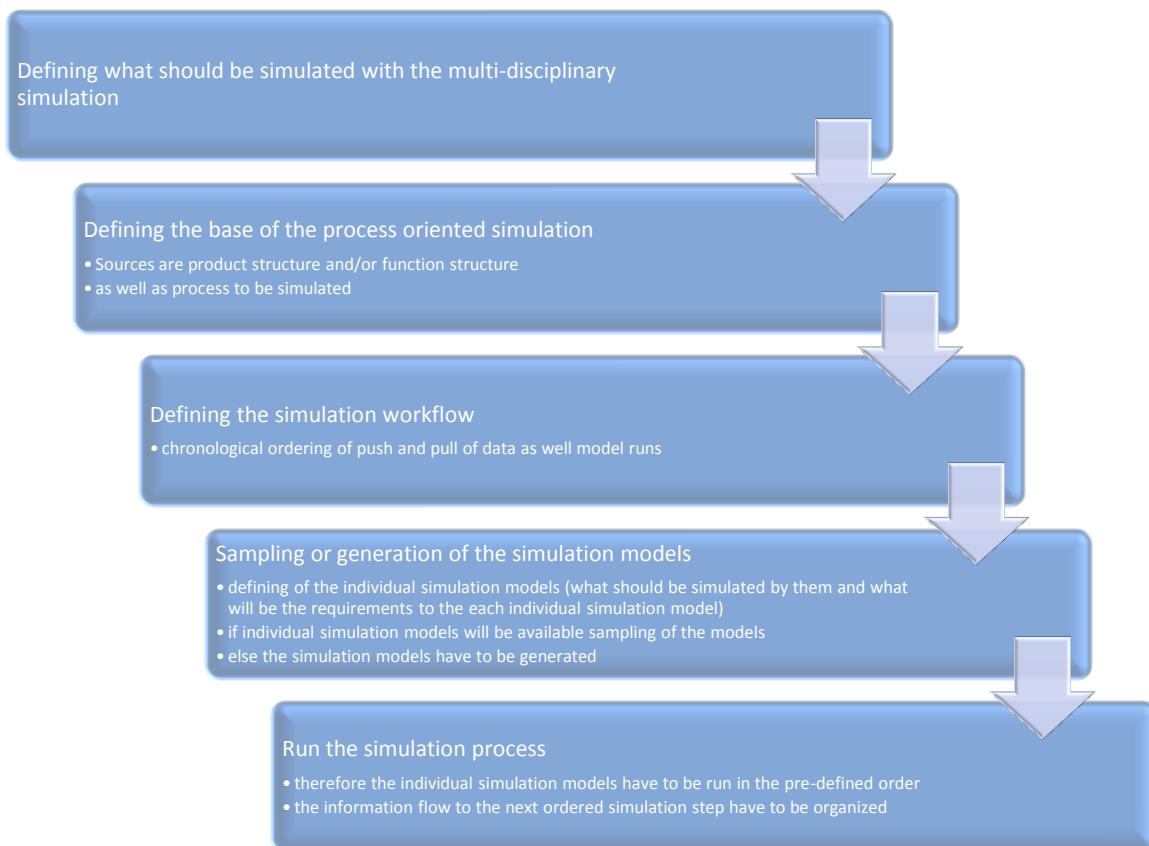
this data for process-oriented simulation processes. In the following, the process-oriented multi-disciplinary simulation of a system-simulation process will be discussed to achieve a greater understanding (see Figure 6-8). The process-oriented multi-disciplinary simulation process is in contrast to the system-oriented multi-disciplinary simulation process discussed in Section 6.1.1 and shown in Figure 6-1.

After the generation of individual simulation models, the information flow of the simulation process must be organised. This means that the different and individual simulation models receive the required input data. Therefore, the data has to be provided to run a process-oriented simulation based on inputs of a previous simulation, for example. This is different to the system-simulation process where the core technology was the co-simulation. A system-simulation can be part of the process-oriented simulation process. With the achievement of (simulation) models and the data information flow between them, a serially structured ordering of the simulation workflow can be achieved. This is called the simulation process. The simulation process can also consist of parallel branches such as in Case 2. The simulation process can be automated. This requires an active management of the data flow; the execution of the individual simulation models embedded in the simulation process. Each individual simulation model is run with its own solver properties.

So, such data flow and simulation execution of process-oriented simulation processes require following functions:

- Transportation of previously generated results from a previous action (maybe simulation) to the following action (maybe simulation), for example, supporting the result file push and pull from a thermal analysis to a structural analysis, as discussed in Cases 1 and 2. This is called “data run”.
- Support or (possibly automated) execution of simulation solve runs, for example, directly opening a simulation that provides the required source files in the authoring or solving tool. Another example could be the opening of the structural analysis with the required thermal result files for execution, as discussed in Cases 1 and 2. This is called “simulation run”.
- Support or (possibly automated) execution of processes such as analysis of results and generation of changed parameters or matrices, for example, providing data of previous simulations; actions like result files to run an analysis; interpretation of the data either in a self-made script or using interpretation software and the retrieval of data produced by the analysis or interpretation. This can also be a data extraction from simulation results, for example to deform geometry like in Case 1. This is called “interpretation run”.
- Support or (possibly automated) execution of processes to change something based on sources, for example, to change parameters, expressions or to run scripts in other tools by self-made scripts or process software, in order to realise geometrical changes or parameter changes, as discussed in Case 1. This is call “change run”.

Each simulation solver produces simulation results or protocols. The review of these simulation results or protocols is completed in the post-process of each individual simulation model. The input of the pre-solved simulation models is also stored as the input in the individual simulation models. So, these simulation models can be handled in the pre-process and in the post-process of the individual simulation model as an independent simulation model. In comparison with a system-oriented simulation process, the simulation models are interacting with a co-simulation. The parallel system-simulation process is shown in Figure 6-8.



**Figure 6-8 Process-oriented Simulation Process**

In Section 6.2.2, an approach to combine the process-oriented simulation process with the system-simulation structure will be discussed.

### 6.2.2 THE PROCESS-ORIENTED SIMULATION APPROACH

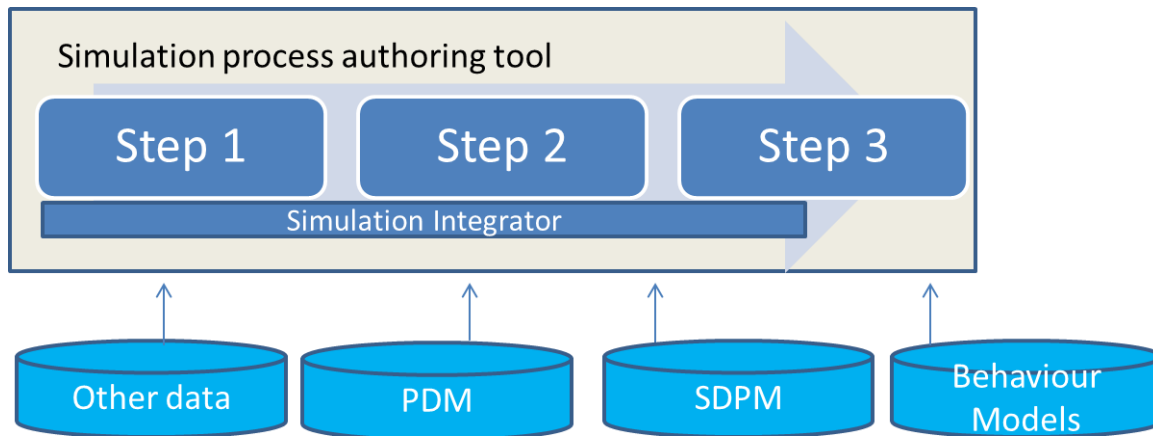
It would be ideal if the system-simulation and the process-oriented simulation processes were implemented in the PLM environment. The architecture of the multi-disciplinary simulation data management should be focused on the system-simulation process. However,

the generated system-simulation models should also be useable for a process-oriented simulation process. Additionally, process-oriented simulation processes are also used to make product optimizations based on CAE results. Commercial products supporting simulation processes are available on the market. Some of these tools include restricted simulation integrators to integrate system-oriented simulations.

The integration of such a tool into simulation data management can provide a solution for the process-oriented simulation process support. Simulation process tools are used in a commercial way and can be prepared to use system-simulation instead of individual simulation models. The data management is not considered to be complex enough to justify a research project. So, there will be no focus on process-oriented simulation processes in the research project.

However, an architectural approach to achieve simulations combining system- and process-orientation could be worked out in the research project. Cases 1 and 2 are, and Case 4 includes, process-oriented simulation processes. The implementation of the optimization of system-simulation models requires the interaction with process-oriented simulation processes. Such an approach can be achieved by using a closed system-simulation as simulation model in a process-oriented simulation.

Simulation process authoring tools such as Isight (Dassault Systems, 2012) are commercial products designed to support, manage and generate simulation processes (Wenzel, Gondhalekar, Balachandra, Guenov & Nunez, 2010). On the whole, the simulation process authoring tools are flexible in the usage of the simulation models taking part in the modelled simulation process. Special functionalities such as parameter changes can limit this flexibility. Such limitations can be observed in the case of optimization processes. Nevertheless, the simulation models are re-used as individual simulation models. In some cases, multiple individual simulation models can be combined with simulation process authoring tools using a provided simulation integrator (Dassault Systems, 2012). However, the provided simulation integrator of a simulation process authoring tool is restricted. Such an approach is shown in Figure 6-9.



**Figure 6-9 Existing Approach of Simulation Process Authoring Tools and Simulation Integrators**

With the determination of system-simulation structure and process-oriented simulation processes, this existing monolithic approach can be improved. Therefore, the new approach can be used to determine between:

- system-simulation structures representing virtual systems that consist of multiple sub-simulation models and
- process-oriented simulation process representing a chronological process of dataflow and model execution (also possible with parallel branches of the process).

An architectural approach can combine system-simulation structures with process-oriented simulation processes. With such an architectural approach, the following improvements can be achieved:

- The process can be better reviewed and handled. This helps to administer the work.
- The system-simulation can be executed externally from the simulation process authoring tool. So, the system simulations are independent from the simulation process authoring tool. The restrictions of co-simulations by simulation process authoring tools are negligible because the simulation integrator is external and independent of the simulation process authoring tool.
- So, the responsibility of the simulation can be assigned to one reliable expert for simulation integration (taking care of the system-simulation) and another for simulation processes (taking care of the process-oriented simulation). Both experts require specific knowledge about their work. The separate handling of such expert knowledge is more effective and realistic than a shared handling.

The improved architectural approach also provides the implementation of models from multiple sources into the simulation process. Additionally, this provides the implementation of a fully configured system-simulation model with a simulation process authoring tool independent simulation integrator. So, the simulation process authoring tool has to handle

the system-simulation model as an individual simulation model. If the system-simulation model is executed, the new framework has to handle the system-simulation work. This approach is monitored in Figure 6-10.

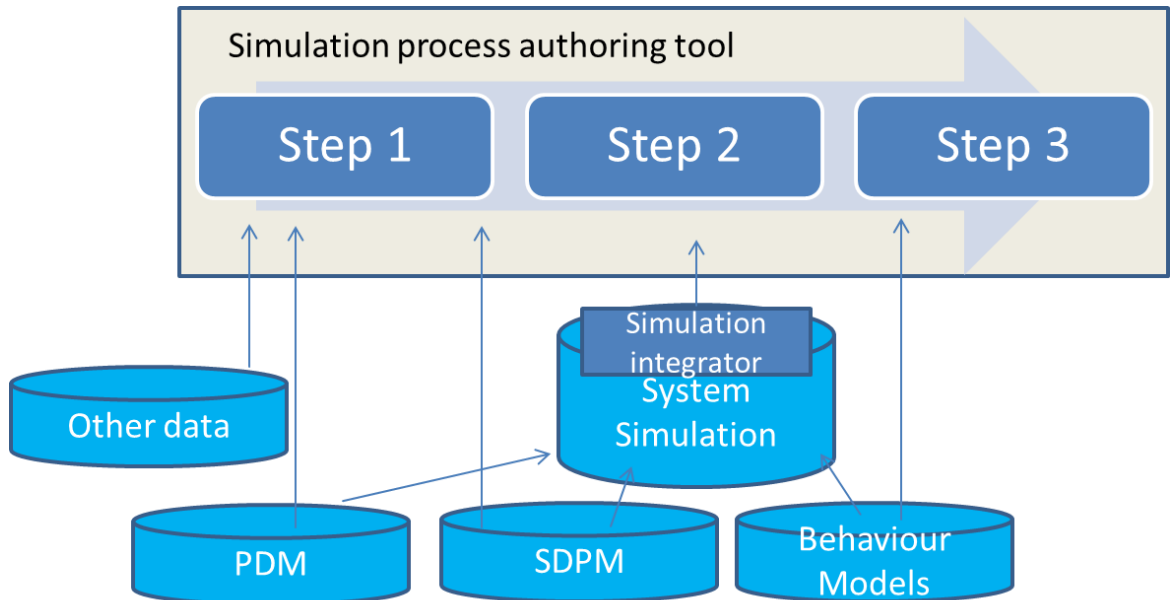


Figure 6-10 Improved Approach of Simulation Process Authoring Tools and Simulation Integrator

The above has been discussed as an architectural approach to support and manage process-oriented simulation processes interacting with system-simulations. The process-oriented simulation processes are not the focus of the research. Nevertheless, a small excursion around some technical details that can achieve such an approach, will help to understand the combination capability of PLM technology.

### 6.2.3 TECHNICAL DETAILING OF THE APPROACH

Four main functions of process-oriented simulation processes were presented in Section 6.2.1. The four functions: data flow, simulation run, interpretation and analysis, and change loops should be provided by a user interface embedded in TEAMCENTER. This should be a user interface for process-oriented simulation processes. In a user interface, similar to MS Visio, the lines represent the data flow and the boxes represent the simulation run, interpretation and change run. Each function and its representation are discussed in the following:

#### Data run line:

The data run line has a starting point, an end point and a direction. The starting point shows where the data is collected and the end point shows where the data is provided. The line should also include information about what the data is (file, parameter, etc.) and dependent upon that, the specification of the data file and dataset as well as parameter name and type.

The push and pull of data can be achieved by linking the source dataset (under the source item) to the end item as a linked dataset. This dataset presents data to each item to which it is linked. So, the dataset linked to each item that requires the data.

The data of the data run line should be managed as a dataset under a simulation process item.

#### Simulation run box:

The simulation run box can have input and output ports. Input ports receive data from a data run. The output ports provide data for a data run. The simulation model can include predefined data types and data names. Therefore, the input and the output ports should include the definition of the kind of information (file, parameter, etc.) and dependent upon that, the specification of the data file and dataset, parameter name and type. This can be used to rename, for example, a parameter or a file, so there is no necessity to make changes in the simulation models. The simulation run box should provide all necessary data. Maybe, it should also open the simulation in the simulation authoring tool. In the case of an automated run of a simulation, the simulation run box can be automated by using the simulation configuration tool (included in TEAMCENTER for Simulation). This configuration of the automated simulation run can be re-used in the simulation run box. Therefore, the simulation run box has to call the configured simulation run based on the item or dataset that should be used. The required item or database also needs to be defined in the simulation run box. The simulation run box should also be stored as a dataset under a simulation process item.

#### Interpretation run box:

The interpretation run box requires an input port. This input port has to deliver the required data for the interpretation. The interpretation run box should include the interpretation routine and required files. This data should be stored under an item as a dataset. In the case of an automated run, the interpretation run box should execute the interpretation run. This can also be an execution routine that can be defined by using the simulation configuration

tool. Therefore, the interpretation run box should be able to execute the routine predefined in the simulation configuration tool. The interpretation can produce output data which should also be provided for re-use as an output port in the interpretation run box.

The interpretation routine can include predefined data types and data names. Therefore, the input and the output ports should include the definition of the information category (file, parameter, etc.) and its dependents, specification of the data file and dataset as well as parameter name and type. This can be used to rename, for example, parameters or files and reduce the necessity to change anything during the interpretation routine.

#### Change run box:

The change run box is similar to the interpretation run box. The difference is that the change run box includes a definition of the object that should be changed during a simulation process. This object should be a dataset of an item. Mostly, this dataset is opened and parameters or expressions are changed by using the authoring tool. The change run box should provide all necessary data and, if possible, open the change object in the authoring tool. The input port should provide information about any changed parameters or data.

In the case of an automated run, the change run box should execute the change run. This can also be a routine that can be defined by the simulation configuration tool. So, the change run box should be able to execute a routine predefined by the simulation configuration tool. This routine can provide output data. However, the required output data should also be provided for re-use as an output port in the change run box.

The change routine can include predefined data types and data names. So, the input and the output port should include the definition of information category (file, parameter, etc.) and its dependents, the specification of the data file and dataset as well as parameter name and type. These can be used to rename, for example, parameters or files and reduce the necessity to change anything in the interpretation routine.

Most functions described previously, can be achieved with TEAMCENTER. However, TEAMCENTER does not provide ordering and managing of the simulation process workflow, i.e. the ability to support and manage process-oriented simulation processes requires a specialised ordering of functionalities to represent the simulation process workflow. Therefore, a user interface is required. In addition to this, an item and dataset are required to manage the elements representing the simulation processes workflow.



The detailing of an architectural approach to support process-oriented simulation processes does not produce new knowledge. Commercial software tools are available on the market and provide approaches. An approach, or an improved approach, of supporting process-oriented simulation process is not the focus of this research project. Section 6.3 will discuss the collaboration of the system-describing structures.

## 6.3 COLLABORATION OF THE SYSTEM-DESCRIBING STRUCTURES

Traditionally, products are described by product data, and functions by functional data. Both represent the system. Similarly, the system-simulation structure represents the system. In the following, these descriptions will be discussed.

### 6.3.1 ANALYSIS OF THE CASES

In Cases 1 and 2, a system-simulation structure is not required. Nevertheless, even in Case 1, the product structure is used to order the simulation models and process. The product structure includes the ordering of the CAD models in assemblies and individual parts.

In Cases 3 and 4, the product structure plays an important role in the simulation structure generation. Here, most simulation was carried out in geometry-oriented way. The geometrics were stored in CAD models. The CAD models were organised by the product structure. Nevertheless, in Cases 3 and 4, EBS models were integrated into the multi-disciplinary simulation. The base of the EBS simulation models were functions. The function hierarchy was organised by function structures.

So, manual sampling of the required simulation models is based on system-describing structures such as product and function. Then, the subsystem simulation models are coupled or merged. This coupling and merging generates the system simulation. Each of the coupled or merged simulation models simulates a specific component or subsystem of the system. This subsystem or component can be based on a specific functionality, a specific assembly or part of the system. Examples of a system simulation based on a system architecture is shown in Figure 5-27. The figure shows which system component is represented by the simulation model and which merging hierarchy is used to generate the system-simulation model. Another example is represented in Section 5.5.5.

So, the product structure represents the system from the product view, and the function structure from the function view. The system-simulation structure represents the system from the simulation view. In summary, there are three system-describing structures. These system-describing structures are:

- the system-simulation structure
- the product structure and
- the function structure

In Section 6.3.2, an approach of interaction between the system-describing structures will be discussed.

### 6.3.2 THE STRUCTURE RELATIONSHIP APPROACH

The generation of a system-simulation starts with the idea to generate a simulation of a product or system. The need for individual simulation models is driven by the product or system. Sampling of the simulation models is done by identifying the simulation models belonging to a product or function. The relationships are often not managed effectively and the sampling is complex, incomplete and incorrect. Mostly, the source of the system-simulation structure is not traceable.

A traceability of sub-ordered simulation models within a system-simulation structure was achieved by the approach presented in Section 6.1. However, additional improvements have to be achieved to make the system-simulation traceable.

Based on the product structure, the function structure or a combination of both, the system-simulation structure can be (pre-) defined. So, based on the product or function structure, the simulation models, sub-ordered to the system-simulation structure, can be sampled. The sampled simulation models are arranged in the system-simulation structure. This means, existing simulation models can be re-used or be placeholders for required simulation models. This requires the identification of simulation models based on the source data such as function or product items. Existing models can be checked for usability. In addition, a new simulation model has to be generated or an existing one has to be changed. In the case of changing an existing simulation model or generating a new simulation model, it should be aligned to the source product or function that should be represented by the simulation model. The source-products or source-functions are sub-ordered to the product or function structure providing the base for the system-simulation.

So, it is necessary to link the system-simulation structure and the sample simulations to the system-describing data sources (product, function structure). However, the product and functional sources are influenced throughout the lifecycle. For example, changes of data can appear during the development process. So, the data has to be linked to each other to make them traceable and show dependencies. This new approach is shown in Figure 6-11.

In contrast, existing systems such as LMS IMAGINE.Lab SysDM and LMS IMAGINE.Lab System Synthesis provide the possibility to import (and duplicate) function sources. With the duplication of the function structure, synchronisation and authenticity checks are not

possible. Additionally, the data is duplicated and managed multiple times. This old approach is shown also in Figure 6-11.

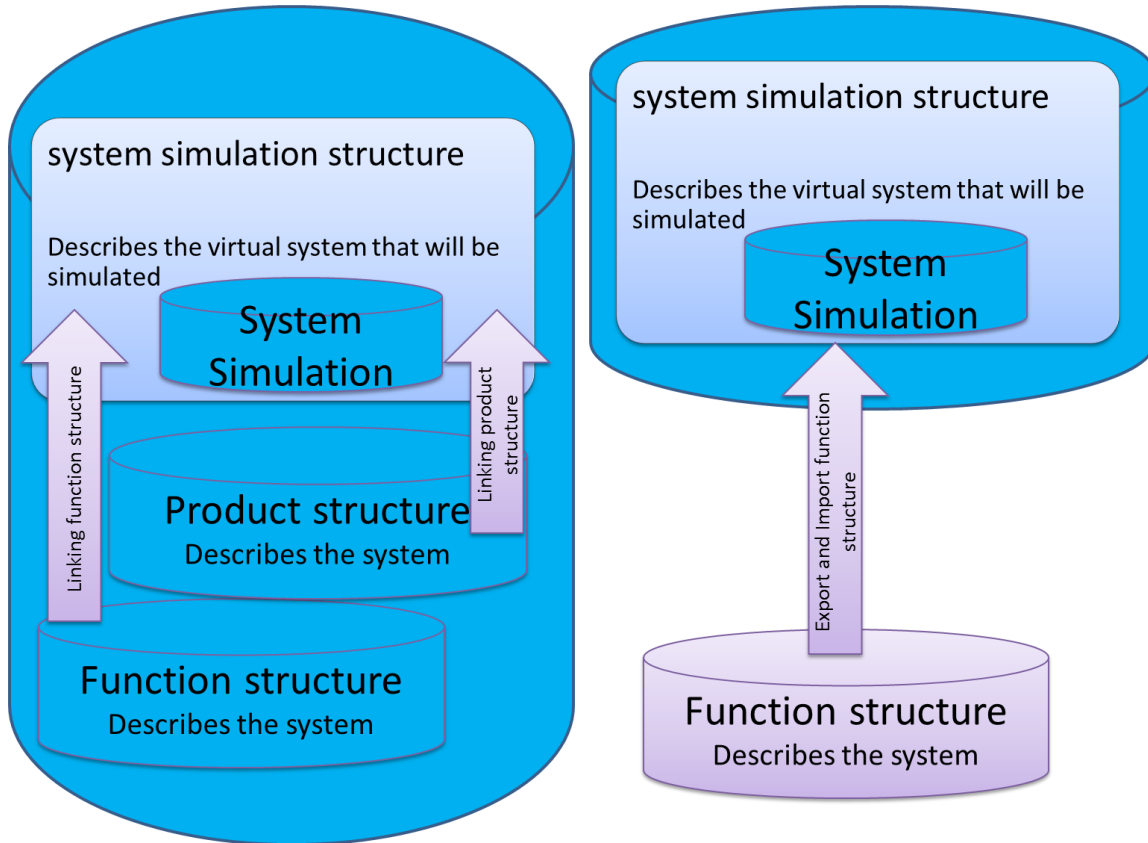


Figure 6-11 left a) New Approach – right b) Old Approach to Link System-Describing Structures

With such a new architectural approach, relationships between the metadata can be generated for representing the dependencies of the data. These relationships can be used to make sure that:

- the unique traceability of the simulation structure to the sources is documented. So, the data has to be managed in one and/ or interacting databases.
- The data duplication is not required anymore due to the linking and relating of the metadata.
- the relationships can also be used to check the stage of the source data such as to verify the authenticity of the data. If they are not authentic, the system-simulation can be reworked.
- the check can be used to synchronise the system-describing structures (system-simulation, product and function structure). So, influences between system-describing structures can be more easily identifiable and reworked.
- the check and synchronization of data supports working with changing data. This helps the administrative work during the lifecycle and development process.

- this synchronization can also be used to derive simulation structures from other system-describing structures.

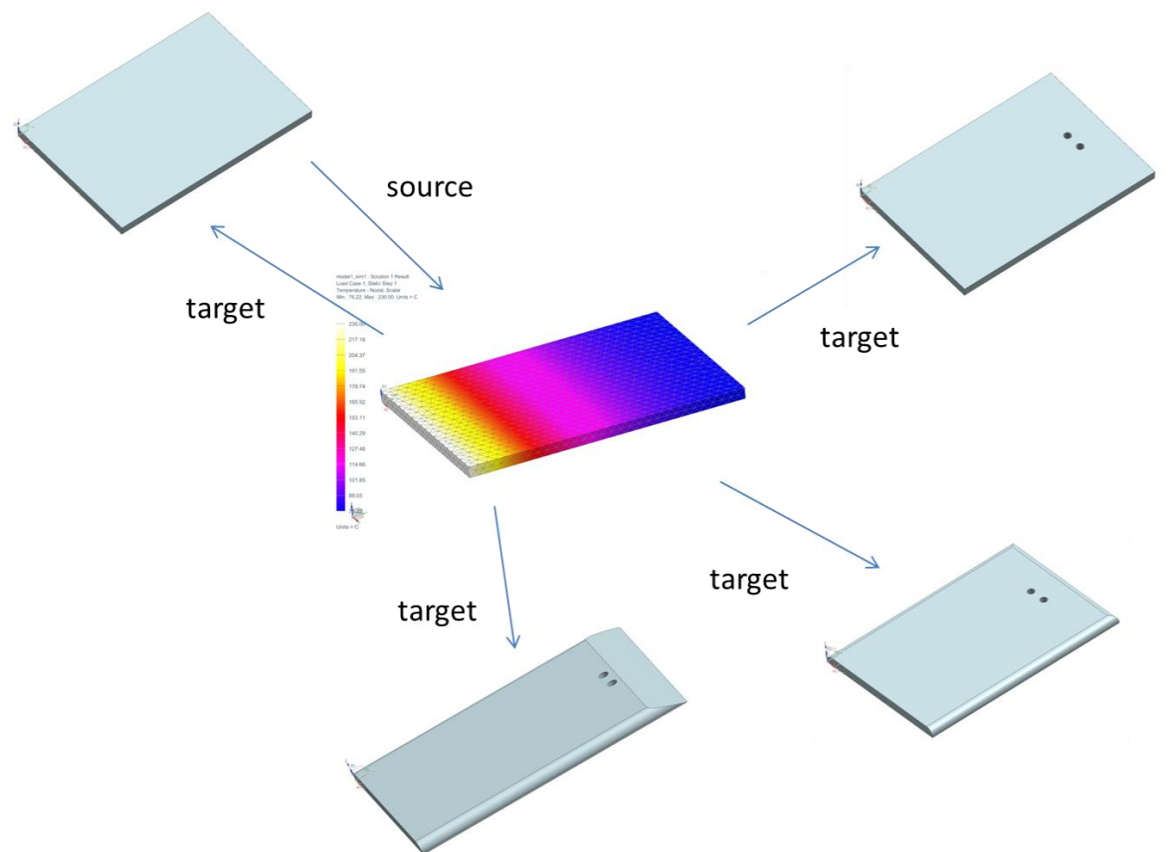
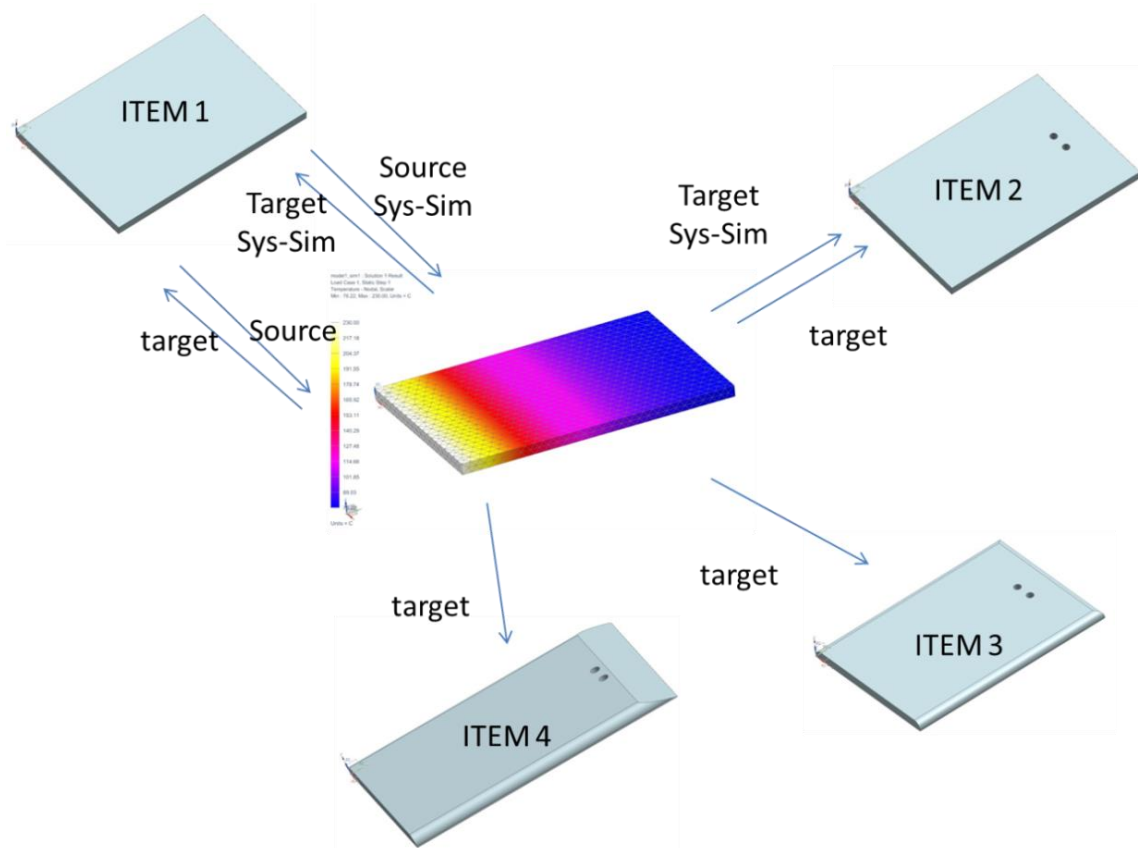


Figure 6-12 Source and Target Relationships

However, it is not easy to achieve a unique traceability. There are differences between sources of the simulation and represented products or functions of the simulation. A simulation can only be based on one source, but it can represent multiple products or functions. For example, a thermal simulation of a steel sheet is based on one source but can represent multiple other steel sheets with negligible features. The example is shown in Figure 6-12. This example is a single model example, but it is the same for the system-simulation structure and all sub-ordered simulation models. In order to address the derivation of sources and representations, two kinds of relationships have to be implemented in the new framework:

- A source relationship relating to the source of the simulation model, as well as
- A target relationship relating to the represented products or functions.



**Figure 6-13 Target and Source Unique Relation of the Simulation Model for Usability in System-Simulation**

A unique traceability to the simulation sources is achieved with the approach discussed in Section 6.1.2. Nevertheless, this approach is focused on the traceability to the sources of the simulation model. This approach does not achieve a reference of the simulation model to be useable as part of a system-simulation. In order to clarify the usability of simulation models for system-simulations, an improved approach has to be implemented. This improvement consists of two additional relationships called Sys-Sim-relations. The Sys-Sim-relationships relate simulation models that are useable in a system-simulation with each other. The relationships should be different between a source relationship and target relationship. Items 1 to 4 are represented by the simulation model and as a target relationship in Figure 6-13. However, the simulation model is only referenced as useable for system-simulation for Items 1 and 2 because the target Sys-Sim-relationship is used. The difference between source and target relationships was discussed above. The Sys-Sim-relationship should achieve:

- a unique traceability of the simulation model to the source product or function,
- a unique traceability to the represented products and functions (the simulation model is classified as useable in a system-simulation structure.),
- the identification of useable simulation models of products or functions and
- the derivation of system-simulation structures by sampling simulation models.

As far as the author is aware such relationships will not be integrated in other system-simulation-approaches. System simulation data management tools such as LMS IMAGINE.Lab SysDM have not been designed to address the lifecycle of source or target data. Such a detailed differentiation between the relations will not be effective in these system simulation data management tools.

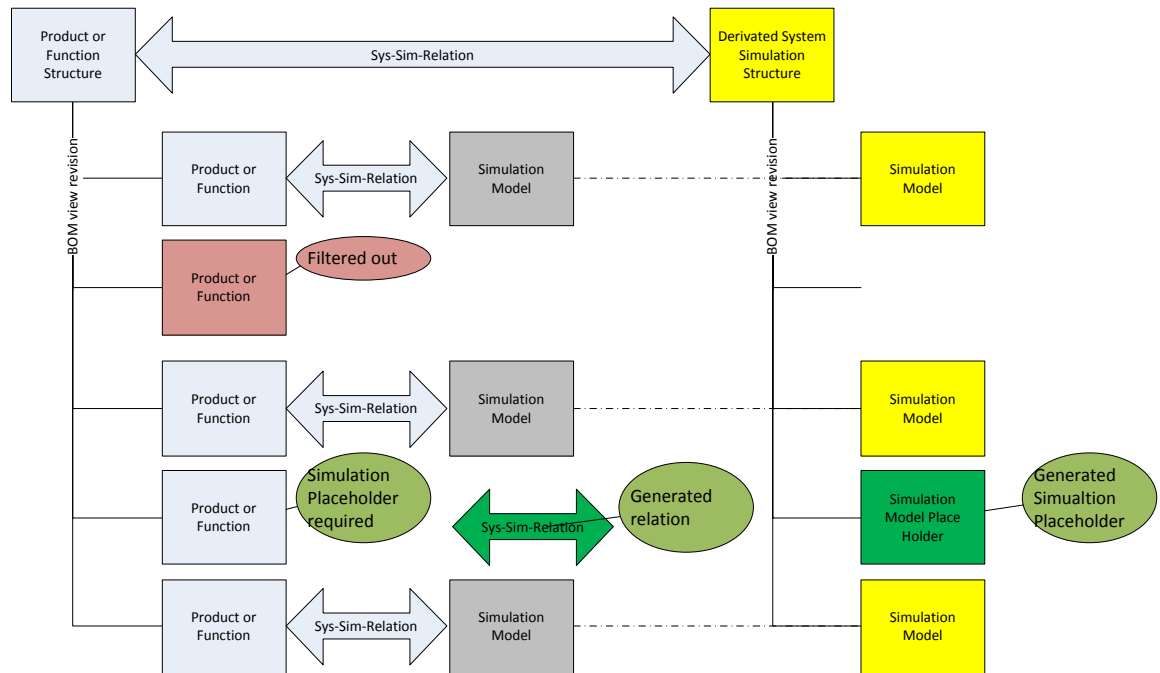


Figure 6-14 System-Simulation Structure Derivation of Product or Functional Structure

The unique traceability of simulation models to the source, and to the represented products or functions makes it possible to derivate system-simulation structures from product or function structures. The derivation follows a pre-definable algorithm. The algorithm has to be user-definable because the derivation of a system-simulation is unique to the use case. Such an algorithm can be generated by defining rules. The algorithm starts with the generation of a new system-simulation structure and the linking to the source product or function structure. A rule should define the kind of relationship (source/target Sys-Sim relationship) between the data. Each sub-ordered product or function of the source product or function structure is identified. With a user-definable filtering rule, redundant functions or products can be identified (metadata filter such as name spelling or parameter values). The filtering can disregard redundant products or functions for on-going work. Each required product or function is checked for the existence of a Sys-Sim relationship. In the following, the Sys-Sim relationship identifies system-simulation-useable simulation models. User-definable rules can check the metadata for the relationship and for the sampled simulation models. Such rules can select or deselect simulation models from the sampling. The structuring of the new system-simulation structure can also be influenced by user-defined rules. These rules should describe the hierarchical sampling of the simulation models. Nevertheless, the relationship (Sys-Sim relationship) between the sub-ordered product or function and the simulation

model, and the structure of the system-simulation structure source product or function structure provides a unique traceability. In the case of missing simulation models, user-defined rules can achieve the generation of placeholders and relationships to the represented product or function. TEAMCENTER for Simulation provides with “Structure Mapping”, a technology to generate algorithms based on user-defined rules for the derivation of CAE model structures from product structures. Functionalities of this technology can be re-used or improved. An example of a system-simulation structure derivation is shown in Figure 6-14.

Hence, the Sys-Sim-relationships provide the possibility:

- Of a unique traceability of data and differences between source and represented data. Therefore, both kinds of data are identifiable.
- To support and interact with all kinds of data (managed as SDPM, PDM or behaviour model, system simulation)
- To reduce the duplication of data by using relationships instead of direct file handling.
- To support data changes during lifecycles (for example during the development process).
- To use traceability and relationships to derive system-simulation structures from product or function structures. Interaction with user definable rules can be used for intelligent system-simulation structure derivation and simulation model sampling.
- To generate algorithms for automated intelligent system-simulation structure derivation.

During the system-simulation structure derivation, placeholders/dataholders for simulation models can be generated. These placeholders can be useful to manage and administer system-simulation projects. Experts to generate the missing simulation models of the placeholders can be identified and commissioned with this work. The placeholders can provide an ideal start for system-simulation projects.



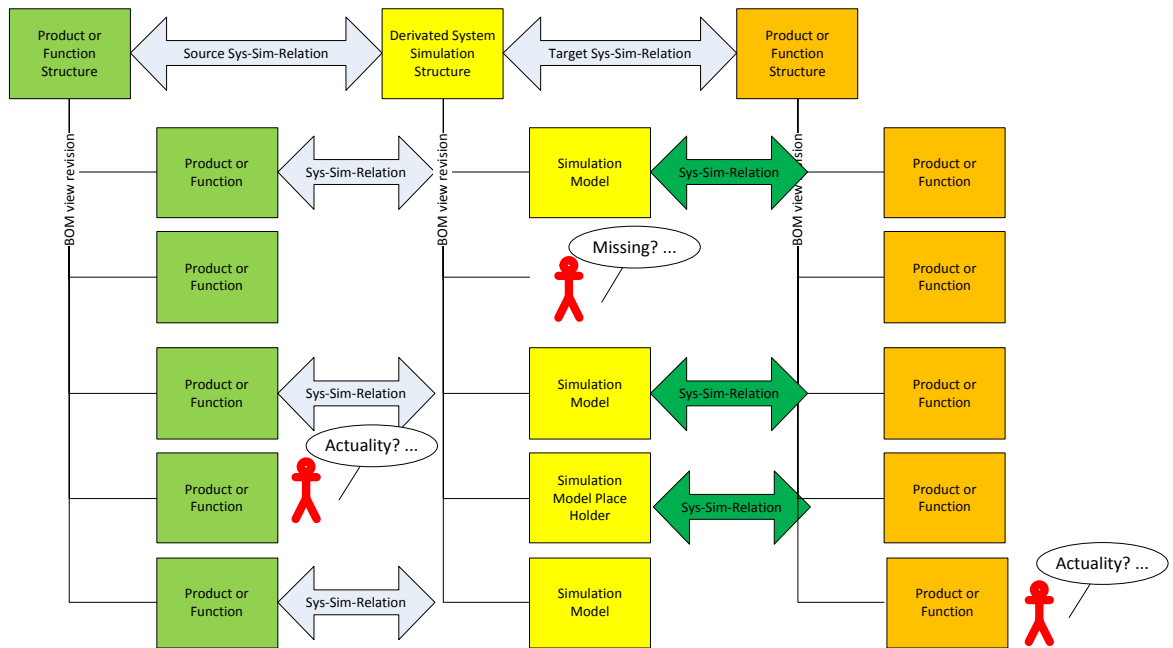


Figure 6-15 Comparison of System-Describing Structures

With the ability to differentiate between source Sys-Sim relations and target Sys-Sim relations, a system-simulation structure can be used relate to product and function structures. The relation to multiple product or function structures can be used to compare the structures. The system-simulation, the product and function structures represent the same system. So, it makes sense to compare them. The comparison can be used to identify missing objects (objects can be simulation models, products or functions and structures) or objects without relationships. This is shown in Figure 6-15. Objects with missing relationships could be objects that differ in the stage of actuality. Such differences can be caused by the development process influencing the lifecycle of objects and with that, the revision stage of objects. More actual revisions can be identified by the foundation technology of TEAMCENTER. With the comparison, the structures will be synchronised. So, an object can be generated or replaced by a more actual one, or a relationship can be added to relate to a more actual object. The synchronization can be done manually. With customer-specific comparison rules, it is possible to automate the synchronization. The CAE-Inspector technology in TEAMCENTER for Simulation provides the possibility to synchronise CAE model structures with product structures. This technology can be re-used or improved for the new framework use case. This technology provides:

- with the Sys-Sim-relationships, the possibility to make system-describing structures dependent on each other.
- the required unique traceability of data for comparison of system-describing structures related to a system-simulation structure.
- the definition of customer-specific comparison rules that make the comparison intelligent.

- intelligent comparisons to synchronise the system-describing structures.
- the possibility to create algorithms and automate the comparison and/or synchronisation.

Section 6.3.3 will discuss the new approach in more detail.

### 6.3.3 TECHNICAL DETAILS OF THE APPROACH

The new approach implements the relationships. The relationships are ordering rules of the simulations and simulation structures, functions or function structures, and products and product structures.

#### I. Ordering Rules to Represent the Dependency of Functions and Function Structures

The understanding of functions and function structures in this context is discussed in Siemens Product Lifecycle Management Software Inc. (2012):

“Functional models illustrate the purposes systems are intended to fulfil. When you design a system or product, you first identify all the functions it should perform and equate each to a functional block. This process creates a functional decomposition of the system or product.

Functional models can be considered as an extension of functional requirements. The form of both is to define the set of inputs, the behaviour, and the outputs of a block of capability. Functional models may provide additional understanding, or a clearer picture, of what the system does.

Decomposition of functional models into finer detail produces parent and child models within a functional architecture. These layers of models describe the functionality of the entire system from the major subsystems to the lowest level subsystems.”

TEAMCENTER manages the functions by using function item revisions. The function item revisions can be ordered to a structure (Siemens Product Lifecycle Management Software Inc., 2012). The structure is built by using BOM view technology. These functions and function structures should be related to the source simulation models. The relationships will order the data, which means the data will be managed under specific objects with specific relationships to each other, to:

- make the source of a simulation model (useable for system-simulation) uniquely traceable to a function item revision. The simulation model is managed under specific item revisions and based on the function or the function structure.

- In such cases, the simulation data requires a relationship to the source function.
  - The function is managed as function item revision.
  - The simulation models can be managed by three different types of item. They are:
    - behaviour item revisions,
    - CAE Model item revisions, and
    - CAE Analysis item revisions.
  - The relation should point from the simulation model to the function source (function item revision). So, the function can be the source for multiple simulation models. However, the simulation model can only be sourced from one function.
  - So, only one source Sys-Sim relationship is allowed for the simulation model.
- The relationship generation is part of the simulation procedure and should be generated by the simulation responsible.
- Actually, available trace link relationships are used in general manner (Siemens Product Lifecycle Management Software Inc., 2012). This means that they do not represent unique traceability. A new relationship, specific for this use case, is required. This new relationship is called source Sys-Sim relationship (there will also be a target Sys-Sim relationship discussed later.)
- make a system-simulation structure uniquely traceable to the source function or function structure.
  - In such cases, the source Sys-Sim relationship should point from the system simulation structure (Sys-Sim item revision) to the source function item revision. So, the function can be the source for multiple system-simulation structures. However, the system-simulation structure can only be sourced from one function.
  - Only one source Sys-Sim relationship is allowed for the system-simulation.
  - The relationship generation is part of the simulation procedure and should be generated by the simulation responsible.

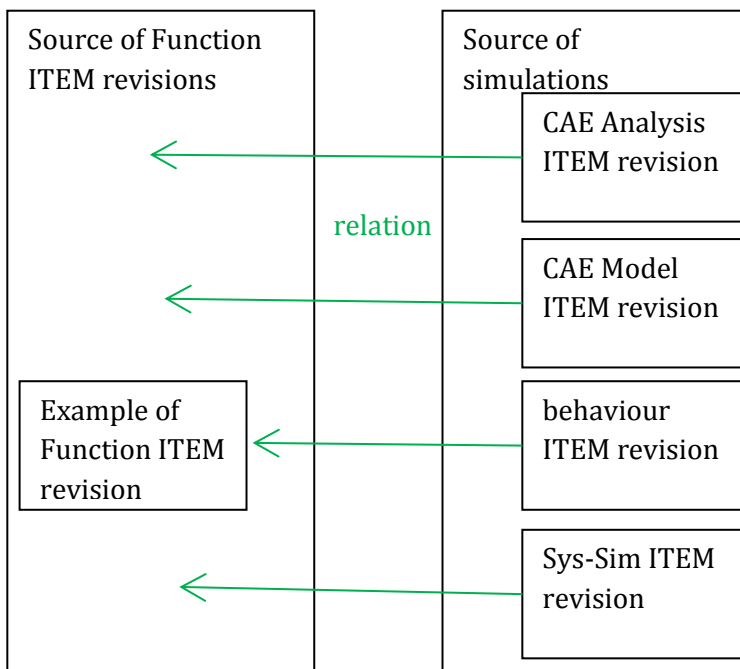


Figure 6-16 Simulation Sources and Function Sources Dependencies

The new source Sys-Sim relationship can be similar to the CAE source relationship of the SDPM provided with TEAMCENTER for Simulation (see Section 4.6). In contrast to this, the new source Sys-Sim relationship should build a unique traceability between the simulation or system-simulation model and the function or function structure. The start point of the source Sys-Sim relationship is a simulation or a system-simulation. The simulation or the system-simulation models can be managed under following item revision types:

- CAE Analysis item revision
- CAE Model item revision
- Behaviour model item revision
- Sys-Sim item revision.

The end point of the source Sys-Sim relationship is the function or function structure managed by following item revision type:

- Function item revision.

This is shown in Figure 6-16.

Nevertheless, a second type of Sys-Sim relationship is required. The source Sys-Sim relationship relates to a source of the simulation or system-simulation. Simulations can represent multiple or other functions differing from the source of the simulation. In order to

monitor the represented function of a simulation useable for system-simulations, the target Sys-Sim relationship will be implemented. Therefore, the relation points from the simulation model to the represented function. That means the Sys-Sim source and the target relationship point from the CAE Analysis item revision, CAE Model item revision, behaviour item revision or the Sys-Sim item revision to the function item revision. So, the target Sys-Sim relationship is similar to the source Sys-Sim relationship and the dialogue of the source Sys-Sim relationship is adoptable by the target Sys-Sim relationship. Only the naming and the numbers of creatable target Sys-Sim relationships differ. A simulation can represent multiple functions. So, multiple target Sys-Sim relationships can point from the same simulation to different functions. This is also shown in Figure 6-17. An exemption could be the CAE Analysis item revision because the source of the CAE Analysis item revision is usually a CAE Model item revision, i.e. the CAE Analysis item revision does not require, in standard use, a Sys-Sim source relation to a Function item revision.

The ability to differentiate between these two kinds of relationships is based on the SDPM application of TEAMCENTER. There are two similar relationships: the CAE source and the CAE target relationship. Both relationships are required to support the lifecycle (during the development process). For example, an additional target relationship is generated if a CAD source is revised or versioned and the existing simulation model can also be re-used for the revised CAD model. In order to document the reusability of the simulation, the new CAD revision or version is linked with an additional target relation to the simulation. So, the reviewer knows that the source of the simulation is the earlier CAD model revision but the newer CAD model revision is also represented by the simulation.

For optimal use of the ordering rules, the relationships have to be reviewable from the start as well as from the end of the relation. The concept of ordering rules for the relationships is also used for the traceability of simulation models to product-describing data.

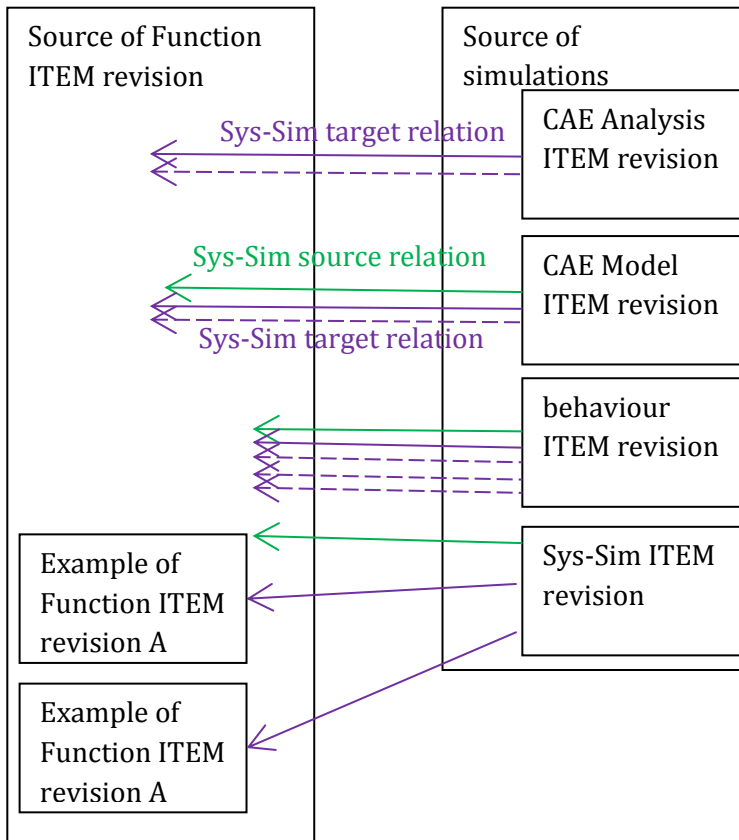


Figure 6-17 Sys-Sim Source and Target Relationship to Functions

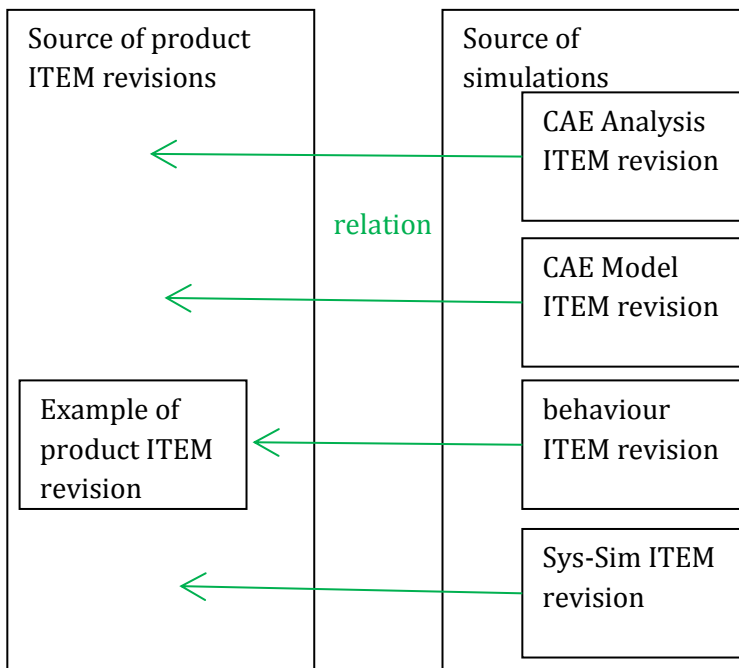
## II. Ordering Rules to Represent the Dependency of Products and Product Structures

An assembly is a structure of multiple products. Unfortunately, this description is flexible and dependent on how the customer uses the product and product structure of a PLM system in this context. A product can also be a supplier assembly and be seen as a single part. Mostly, the product is the smallest part of the assembly that is not detached. It can be a single screw or a supplier assembly. So, a product is dependent on the PLM user's point of view and managed by using item revisions in TEAMCENTER. The item revisions can be linked to a structure (Siemens Product Lifecycle Management Software Inc., 2012). The structure is built by using BOM view technology. These products and product structures should be related to the source or representing simulation models. The relationships will order the data, which means the data will be managed under specific objects with specific relationships to each other, to:

- make the source of a simulation model (useable for system-simulation) uniquely traceable to a product item revision. The simulation model is managed under specific item revisions and based on the product or the product structure.
  - In such cases, the simulation data requires a relationship to the source product.
    - The product is managed as item revision.
    - The simulation models can be managed by three different types of item. They are:
      - behaviour item revisions
      - CAE Model item revisions, and
      - CAE Analysis item revisions
    - The relation should point from the simulation model to the product source (item revision). So, the products can be the source for multiple simulation models. The simulation model can only be sourced from one product.
    - So, only one source Sys-Sim relationship is allowed for the simulation model.
  - The relationship generation is part of the simulation procedure and should be generated by the simulation responsible.
  - Actually, available trace links relationships are used in a general manner (Siemens Product Lifecycle Management Software Inc., 2012). Unfortunately, this means they do not represent unique traceability. A new relationship, specific for this use case, is required. This new relationship is called source Sys-Sim relationship (there will also be a target Sys-Sim relationship discussed later).
- make a system-simulation structure uniquely traceable to the source product or product structure.
  - In such cases, the source Sys-Sim relationship should point from the system simulation structure (Sys-Sim item revision) to the source product item revision. So, the product can be the source for multiple system-simulation structures. However, the system simulation structure can only be sourced from one product.
  - Only one source Sys-Sim relationship is allowed for the system-simulation.
  - The relationship generation is part of the simulation procedure and should be generated by the simulation responsible.
- monitor the represented product or product function by the simulation or system-simulation.
  - The target Sys-Sim relationship relates the simulation to the represented product or product structure.
  - Therefore, the target Sys-Sim relationship points from the simulation model or system-simulation which can be:
    - behaviour item revisions
    - CAE Model item revisions



- CAE Analysis item revisions, or
- Sys-Sim item revision.
- to the product which is
  - item revision
- So, the simulation can represent multiple products. Multiple target Sys-Sim relationships can appear and be required for the same simulation.
- The relationship generation is part of the simulation procedure and should be generated by the simulation responsible.



**Figure 6-18 Simulation Sources and Product Sources Dependencies**

The relationships can be tracked from the simulation or the system-simulation to the product. The start point of the relationships can be a:

- CAE Analysis item revision
- CAE Model item revision
- Behaviour model item revision, or
- Sys-Sim item revision.

The end point of the Sys-Sim relationship is:

- a product item revision

This is shown in Figure 6-18. The relationships differ between source Sys-Sim relationships and target Sys-Sim relationships. The source points to the source of the simulation and the target points to the represented product. This is shown in Figure 6-19.

The ordering rules and relationships are the same as discussed for the functions. So, the ordering rules and relationships of the products and the functions to the simulation models are the same and can be mixed.

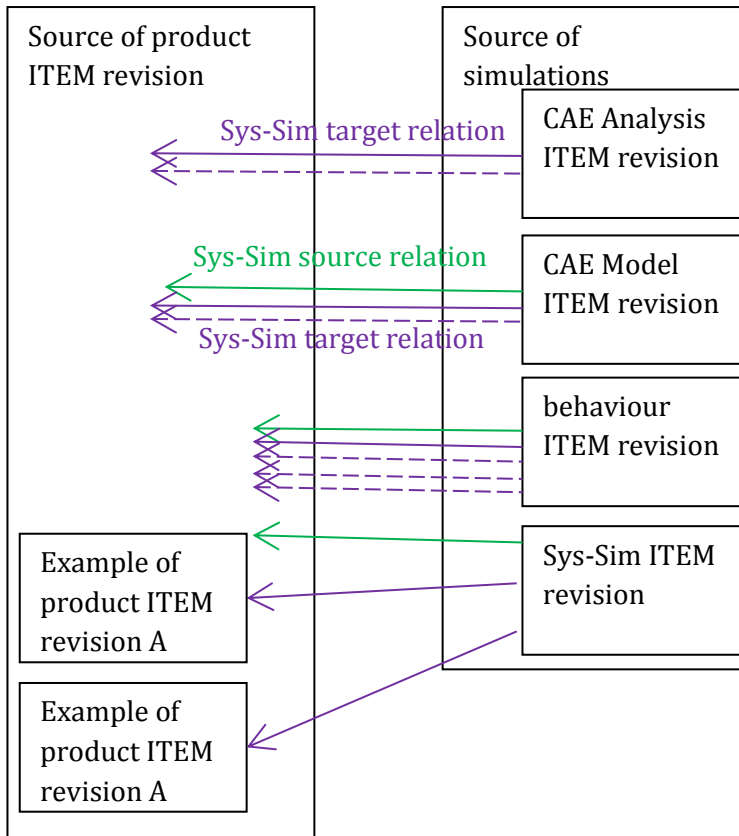


Figure 6-19 Sys-Sim Source and Target Relationship to Functions

In summary, the Sys-Sim source and target relationships are used as ordering rules to represent simulation model dependencies, including the system and behaviour and other simulation models. The Sys-Sim relationships can be used to represent relationships from simulations to functions and products. The Sys-Sim relationships can differ between simulation sources and represented functions or products of the simulation models. These ordering rules can be used for synchronisation and derivation.

### III. Algorithms and Rules for Synchronisation and Derivation

The ordering rules provide unique traceability of the data. This unique traceability can be used to review the data which makes it possible to:

- review structures, such as a system-simulation structure, for sub-ordered objects, such as simulation and system-simulation objects. This review is based on BOM view technology used to assemble objects to the top object. Reviewing this BOM provides information about the sub-assembled objects. BOM view technology is used by TEAMCENTER for the structuring of data (such as for function, product, simulation as well as system-simulation and other data). Sub-assembled objects can sub-assemble additional structures with BOM view technology. However, these structures are also reviewable by the BOM view. Using BOM view technology, the ordering rules provide the possibility to review the structures and identify each object in the structure.
- review dependencies of simulations and sources. Therefore, source Sys-Sim relationships are implemented. These relationships can be reviewed from the source, which is the end of the relation, and from the simulation sight, which is the starting point of the relation. It is only permissible for a simulation model to have one source Sys-Sim relationship. So, the source of the simulation and simulation models based on a product or function can be reviewed.
- review dependencies of simulations and represented objects. Therefore, target Sys-Sim relationships are implemented. This relationship can be reviewed from the product or function, which is the end of the relation, and from the simulation sight, which is the starting point of the relation. It is permissible for a simulation model to have more than one target Sys-Sim relationship. So, the represented object of the simulation and the simulation models representing a product or function can be reviewed.

Thus, the ordering rules include new implemented objects, relations and BOM views to create structures. All these newly implemented objects, relations and BOM structures are based on existing technologies and provided using the underpinning of TEAMCENTER. Using this existing technology to create new ordering rules makes the new ordering rules reviewable by using TEAMCENTER base functionality.

The ability to review the data based on the ordering rules helps:

- to identify dependent objects, such as dependent sources of simulations or dependent simulation models of products.
- to identify the dependency category, such as the difference between source or representation dependencies.

These existing technologies (in the foundation of TEAMCENTER) and the new ordering rules can be used to compare data. The comparison of data is based on data dependency. An intelligent review of data dependencies can achieve the comparison. Such intelligent reviews run in a serial process. For example, the following review rule sets can be adopted to compare a system-simulation structure with a source structure:

- 1) Check if the selected starting object is a system-simulation (Sys-Sim item revision). If not, cancel the process.
- 2) Check if the system simulation structure provides a BOM (Sys-Sim item revision BOM view). If not, cancel the process.
- 3) Review the system-simulation structure BOM view (Sys-Sim item revision BOM view) and identify the assembled objects.
- 4) Review each of the identified objects for the existence of source Sys-Sim relations and identify the source of the objects. Put the source dependency of each simulation object into the memory.
- 5) Check if a source Sys-Sim relationship is available (source Sys-Sim relation). If not cancel, the process.
- 6) Review the source Sys-Sim relationship and identify the source object (source Sys-Sim relation).
- 7) Check if the source object includes a BOM (product or function item revision). If not, cancel the process.
- 8) Review the BOM view of the source object (product or function item revision BOM view) and identify the assembled products or function objects.
- 9) Compare each assembled product or function with the source dependency of the simulation models (assembled under the system-simulation). The source dependency of the simulation models will be available through the data management memory.
- 10) Monitor the comparison of both structures (the system-simulation structure and the source structure) in one user interaction view. In the case of a given dependency, mark/link the source object to the simulation object.

Such a rule set is dependent on the use case. Multiple rule sets can be required. Thus, it is useful for the user to define the rules himself/herself or use predefined rule sets. Such a rule set generates algorithms for comparison.

These kinds of rule sets can be improved by adding filtering rules. Filtering rules can make objects negligible. A filtering rule should be able to check the metadata of identified objects. Based on this check, it can be decided if the object should be followed up or not.

For the synchronisation and derivation of system-simulation structures, additional rules are required. In both cases, it can be required to generate new items and item revisions, relations or existing items and item revisions have to be related or integrated to BOM views. In the case of the synchronisation, it can also be required to remove objects such as item revisions or relationships. For removing objects, a removal rule, and for the generation of objects, a

generation rule should be provided. In addition, a re-use rule is required to re-use objects in a BOM view or to relate them to other objects.

TEAMCENTER for Simulation uses “Structure Mapping” functionality such filtering rules, removal, creation and re-use rules together with existing ordering rules and review technologies. The rule set defined by this structure mapping creates algorithms that execute the rule set. This technology can be re-used or improved to support the new ordering rules of the new framework and the new functionalities for the creation of comparison, synchronisation and derivation algorithms.

## 6.4 INSTITUTION OF AN SYSTEM-SIMULATION RESULT STRUCTURE

A system simulation execution generates system-simulation and a set of single simulation results (McLean, Riddick, & Lee, 2005). The results of a multi-disciplinary simulation are required to be related to the production of a simulation model. In Section 6.4.1, the handling of the results will be discussed.

### 6.4.1 ANALYSIS OF THE CASES

Cases 3 and 4 require a dependency of the individual simulation models to the system-simulation and to the individual simulation model of the FEM and MKS simulation. The dependency to the individual simulation model is required for the post-processing of the individual simulation model results. So, a specific simulation authoring tool is used as a post-processor. Alternatively, a post-processing tool that supports different results of the simulation models is necessary. Unfortunately, post-processing tools are not available for meeting this alternative requirement. An additional argument is that the experts, who created the individual simulation models, work in their used and trained environment.

In Cases 1 and 2, the simulation results can be directly related to the simulation models. This is achieved through standard management of the results. The complexity of the result data management is low because Cases 1 and 2 are process-oriented simulations. In a process-oriented simulation, individual simulation models can be solved in chronological order. The solve run is independent from other simulation solve runs. Only simulation results of the actual simulation solve run of a specific simulation model are generated. The simulation results are directly appended to the simulation model solved in the actual step. So, the simulation result data can be managed and directly related to the responsible simulation model.

Moreover, the result files management of a system-simulation solve run is not easy. In Cases 3 and 4, there are such system-simulation solve runs. Multiple simulation models are solved

in parallel and in co-simulation. So, multiple simulation models are solved at once, which causes the immediate generation and edition of multiple simulation result files. The result data should relate to the system-simulation. It should also be differentiated and related to the responsible individual simulation models in the system-simulation. In Cases 3 and 4, all simulation result files are related to the co-simulation master simulation model. Post-processing of an individual simulation model based on the generated results during a system-simulation run requires manual handling of the data. This manual handling is required because of missing relationships between simulation models produced by an individual simulation model during system-simulation solve runs.

### 6.4.2 THE SYSTEM-SIMULATION RESULT STRUCTURE APPROACH

A system-simulation execution generates system-simulations as well as individual simulation results (McLean, Riddick, & Lee, 2005). These simulation results should be stored and managed dependent on the system-simulation structure. However, the storing of all results in the system-simulation structure does not support the possibility to interpret and analyse them individually. Only the management of the results generated by the simulation middleware can to be directly managed under the level of the system-simulation structure head. This is the result of Cases 3 and 4 analyses in the previous section.

The simulation results generated by individual simulation models in the system-simulation should be managed, based on those simulation models. This supports individual post-processing because the simulation authoring tools of the simulation models provide the possibility of post-processing. They require, therefore, the simulation results. So, a new approach is required. Such an approach should improve the handling of system-simulation results and provide the possibility of organizing the results data dependent on the system-simulation and the simulation models.

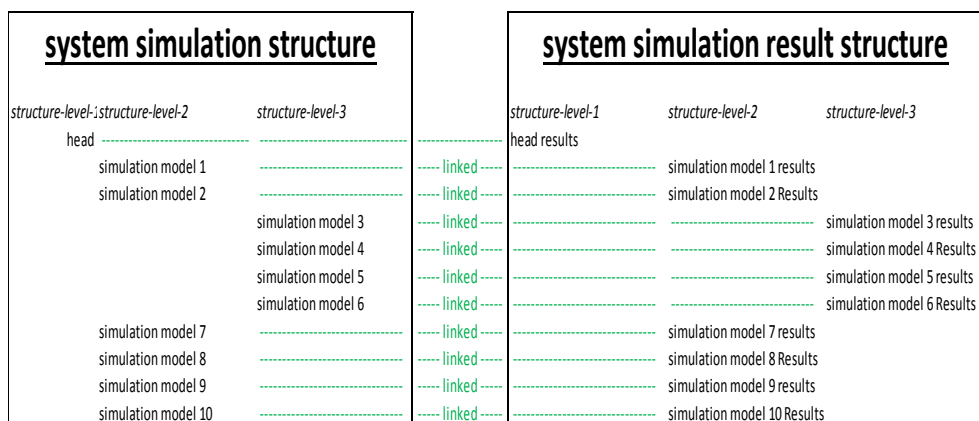


Figure 6-20 New Approach of System-Simulation Result Structure Linking

The new framework implements system-simulation result structures. The system-simulation result structure is structured in a similar way to the system-simulation structure. At the top level of the system-simulation result structure, the result of the simulation middleware should be managed. The structure head should be linked to the system-simulation structure head. Assembled to the system-simulation result head, individual simulations or sub-ordered system-simulations should be managed. Sub ordered result objects should also be linked to the responsible simulation model in the system-simulation structure. This is shown in Figure 6-20. The newly implemented Sys-Sim-Result relationship relates the simulation object to the dependent simulation result object.

These relationships present the dependencies of the individual simulation models to the individual simulation results. The new architectural approach of a system-simulation result structure and the Sys-Sim-Result-relationships can generate the dependencies of simulation and result data. Based on these ordering rules:

- The simulation results can be partitioned into multiple simulation result objects.
- The multiple simulation result objects can be sub-ordered to a simulation result object responsible for the system-simulation.
- Each of these simulation result objects can be related to the responsible simulation model in the system-simulation.

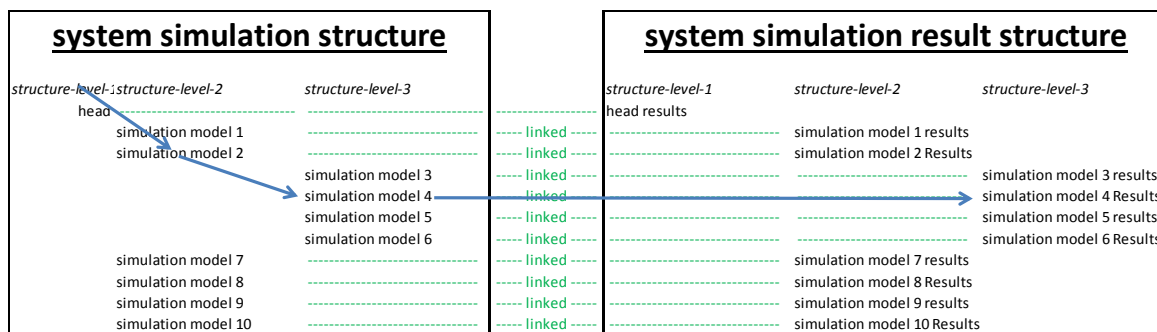


Figure 6-21 Example of Simulation Result Traceability

Users can use these relationships to identify the simulation results of each simulation model. An example can be the identification of simulation results generated by a specific simulation model in a system-simulation. This example can be reviewed in Figure 6-21. Based on the ordering rules:

- A unique traceability of simulation results (generated during a system-simulation) to the responsible simulation model will be given.
- This unique traceability can improve the post-processing of the simulation results. The relation will provide an easy, manual review of the data.
- The review can be automated to provide the right data for post-processing.
- A more common collaboration environment is provided by the ordering rules. They improve the post-processing of system-simulation and provide the result data management dependent to the responsible simulation model and to the responsible

system-simulation. The new approach has the support of multiple simulation experts on multiple simulation data. Such a collaboration environment will improve the post-processing of a system-simulation from a single level to multiple levels of sub-post-processing.

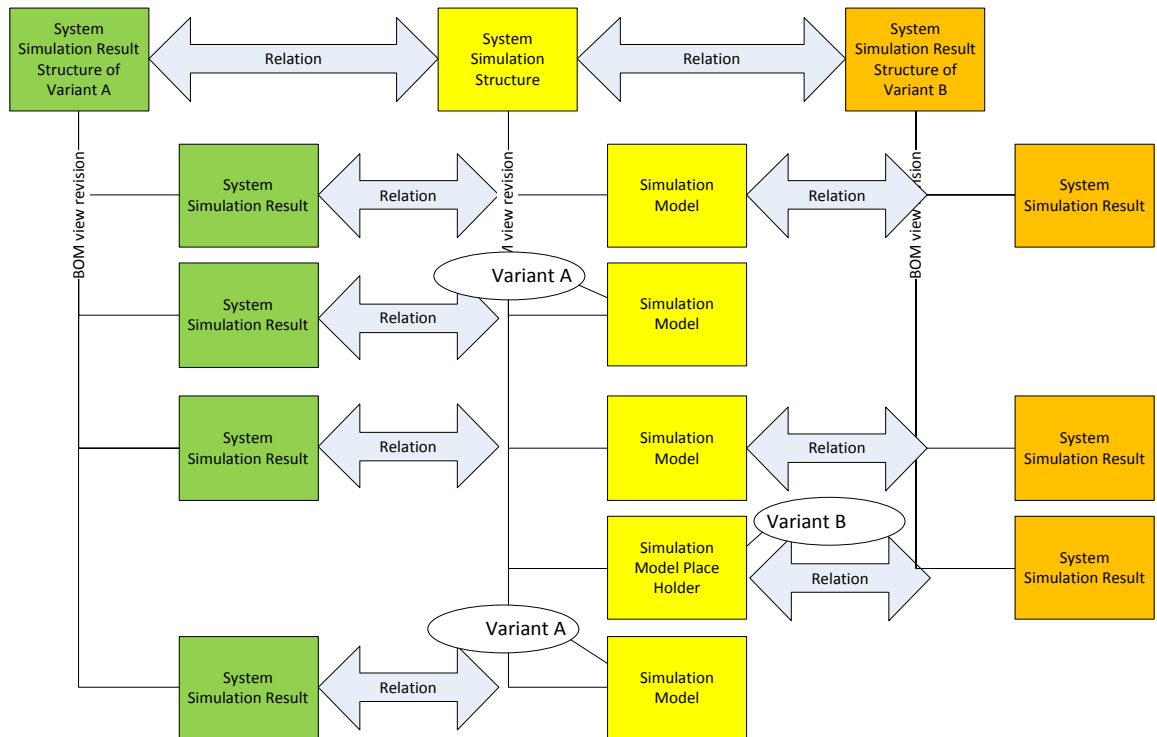


Figure 6-22 Example of System-Simulation Result Management at Multiple System-Simulation Solve Runs

The new approach to system-simulation result management can support the result management of multiple system-simulation solve runs. Multiple system-simulation solve runs can be required in the case of system-simulation structure variants or system-simulation parameter studies. So, multiple system-simulation result objects and multiple system-simulation structures can be generated and related to the responsible simulations in the system-simulation. In the case of variant studies, the system-simulation structure can be a 150% structure. A 150% structure means a structure that includes different variants of the structure. Such variants could include and exclude sub-ordered elements of the structure. So the structure will include more than 100% of the sub-ordered elements. So, such a structure will be called 150% structure. A filtering of the structure could activate or deactivate sub-ordered elements. In the case of the system simulation structure, this means that variant filtering rules activate or deactivate sub-assembled simulation models. Not all assembled simulation models of the system-simulation take part in the system-simulation solve run. This is monitored in Figure 6-22. Based on the ordering rules:



- A unique traceability of simulation results generated during multiple system-simulations is achievable. Multiple system-simulations mean that variant and parameter studies are supported.
- The unique traceability of the multiple simulation results to the multiple responsible simulation models is also achieved.

This system-simulation result management is not identified in the other tools. In the case of system-simulation data management tools, the results are usually managed under the top system-simulation header (see Figure 6-23). This is because such systems are focused on the management of behaviour simulation data. The behaviour simulation data is mostly integrated into one simulation model. This is possible because the simulation models are generated by the same EBS simulation authoring tool or use standard formats such as MODELICA. In some cases of behaviour simulation models, external simulation models are also implemented. Then, the commercial system-simulation management systems have copied this data from external resources. All result files are managed at the system-simulation top object.

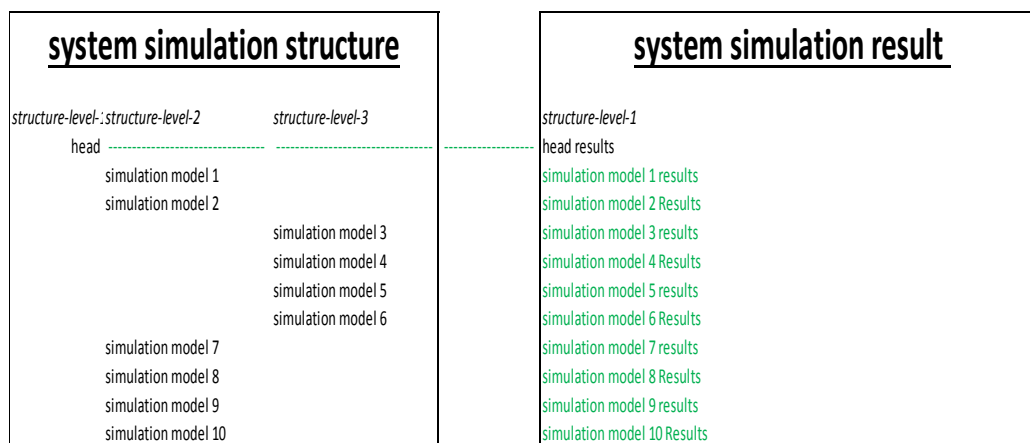


Figure 6-23 Old Approach of System-Simulation Result Structure Linking

Section 6.4.3 will discuss the new approach in more detail.

### 6.4.3 TECHNICAL DETAILING OF THE APPROACH

In the post-processing phase, the result files and result data, generated during the solve run of the system-simulation, have to be managed. The solve run of a system-simulation consists of multiple solve runs of individual and single simulation models integrated into the system-simulation. "Should these files (log-files and result files of the solve run) be managed and

aligned to the individual and single simulation models or to the system simulation model?" The answer is: "Both."

In the case of the simulation data and process management environment of TEAMCENTER, simulation results are managed under a CAE Result item revision (see Section 4.6). However, the management of the data using the CAE Result item revision does not relate it to the system-simulation or to the responsible simulation model either. In order to explain that the results are generated by a system-simulation, a new item revision type called Sys-Sim Result item revision should be instituted. The Sys-Sim Result item revision can be used instead of a CAE Result item revision. The Sys-Sim Result item revision has the same functionality as a CAE Result item revision. It can be linked to the CAE Analysis item revision using the existing CAE Result relationship. The same is done for the CAE Result item revision. However, the naming of the Sys-Sim Result item revision refers to the fact that the results are generated during a system-simulation solve run and not by an individual simulation solve run. The Sys-Sim Result item revision can also be used in the case of behaviour simulation models or product item revisions. So, the Sys-Sim Result item revision is related by the CAE Result relationship to the responsible simulation model.

The CAE Result relationship points from the Sys-Sim Result item revision to the source simulation model. A source simulation model can be a CAE Analysis item revision, CAE Model item revision, behaviour item revision or product item revision. For the new framework and ordering rules, the existing CAE Result relationships have to be improved to support all these named items.

Under multiple Sys-Sim item revisions, the result files, generated during a system-simulation solve run for each individual simulation and also for the system-simulation, are managed. The traceability of the Sys-Sim Result item revision to the responsible simulation is provided by the CAE Result relationship. However, the traceability of the multiple Sys-Sim Result item revisions to the system simulation result structure is missing. The system-simulation result structure is also managed as a Sys-Sim Result item revision and related by a CAE Result relationship to the system simulation structure. So a Sys-Sim Result item revision could be generated for each simulation object beginning at the top level of the system-simulation structure. The generated multiple Sys-Sim Result item revisions have to be hierarchically structured.

The hierarchical structuring of the system-simulation result structure is achieved with BOM view technology. BOM view technology assembles Sys-Sim Result item revisions of the subsequent lower level to a top Sys-Sim Result item revision. This can be repeated at sub-ordered Sys-Sim Result item revisions. A hierarchical structure is realised.

This achieves traceability of sub-ordered Sys-Sim Result item revisions hierarchically to a top Sys-Sim Result item revision. The top Sys-Sim Result item revision is traced by a CAE Result relationship to the system-simulation structure (Sys-Sim item revision). In addition, the other Sys-Sim Result item revisions are related by the CAE Result relationship to respective simulation models. So, new ordering rules including the Sys-Sim Result item revisions,

together with the BOM technology and the relating of each Sys-Sim Result item revision by the CAE Result relationship, are implemented into the new framework. These new ordering rules provide the dependency of the results to the system-simulation and to their respective simulations. This is shown in Figure 6-24.

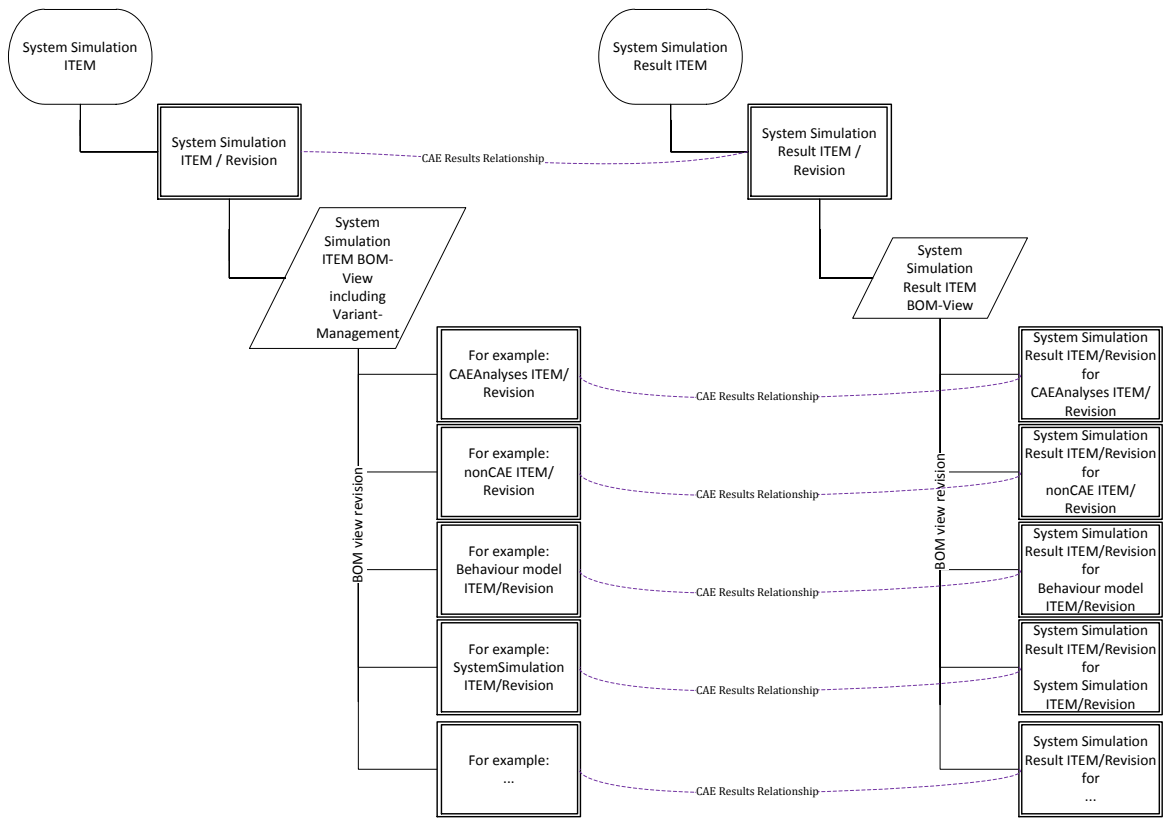


Figure 6-24 Ordering Rules for Sys-Sim ITEM Revision and Sys-Sim Result ITEM Revision

### 6.5 THE ARCHITECTURAL CONCEPT OF THE NEW FRAMEWORK

The design of the new framework for supporting and managing multi-disciplinary simulation data is influenced by system-simulation processes and the implementation into the PLM environment. The new framework should be open to multiple simulation disciplines, such as physical disciplines, as well as vendors and inter-disciplinary cooperation (Tian, Yan, Parkin, & Jackson , 2008). These factors have to be solved by a common approach to the new framework. Therefore, an architectural concept of the new framework has been proposed. This architectural concept does not include application generation. In this Section, the new framework and the architectural concept approach will be discussed.

### 6.5.1 ANALYSIS OF THE CASES

The discussion and Cases 3 and 4 shows that there can be two kinds of sources for simulation models:

- Function and requirement data, such as RFLP data, originating from the system engineering approach. This base is in light green in Figure 6-26.
- The product data mainly handled in PDM. This base is in dark green in Figure 6-26.

These sources can provide the basis for simulation data. In all case studies, it is obvious that PDM can provide CAD models as sources for FEM or MBS simulation models. The same applies in the case of CFD (computational fluid dynamic simulation). In some cases, PDM manages the simulation data directly, for example, in Case 3, the rigid body simulation is directly ordered to the PDM.

In Cases 3 (see Figure 5-30) and 4 (see Figure 5-51), the logical simulation represents functions or function structures. In Case example 3, the logical simulation represents a functional structure of a software and controller system. However, functions can also describe mechanical or electrical systems such as discussed in Case 4 (see sub-section 5.5.3 I Logical Simulation Model of Case 4). These functions can be represented by behaviour simulation models (see Section 4.8.1). This way, behaviour models can be generated earlier in the development process. Therefore, the behaviour models can be used early on in the development process for mechatronic concept optimization. This can appear before the detailed discipline-oriented development departments begin their work. So, system engineering data such as requirements, functions, logical and product data provide a base for simulation data.

The management of the simulation data was achieved in these case examples with TEAMCENTER. TEAMCENTER provides three possibilities:

- A more common solution is the simulation data and process management of TEAMCENTER. This solution is mainly used for FEM, CFD or MKS simulation data, but it is an open approach for all kinds of simulations. This base is in dark blue in Figure 6-26.
- A more specific solution is provided with the behaviour model management. This base is in light blue in Figure 6-26.
- A direct management of simulation data as a dataset based on PDM. This base is in dark green in Figure 6-26.

In summary, the above discussion explains the five base boxes of Figure 6-26. The top boxes couple different simulation models.

The coupling of different simulation models is the challenge of a simulation middleware (McLean, Riddick, & Lee, 2005). A simulation middleware is not focused on generating simulation models. It is not an authoring tool for building or solving simulation models. The

focus of a simulation middleware is the organization and management of the co-simulation (McLean, Riddick, & Lee, 2005). Benedikt, Zehetner, Watzenig & Bernasch (2012) discussed:

“The purpose of a co-simulation platform is to take into account the complex interactions between different simulation models in a proper and correct manner. The platform must therefore enable the precise cooperation of several simulation tools.”

Such simulation middleware examples are ICOS from the virtual vehicle company in Graz or the middleware TISC Suite from the TLK-Thermo GmbH in Braunschweig. This software tool provides the opportunity to be independent from the used simulation solvers in a co-simulation. This offers higher flexibility to create co-simulation models caused by the openness. Simulation middleware is also able to integrate correction algorithms into a co-simulation in order to reduce errors during co-simulation (Scharff, Kaiser, Tegethoff, & Huhn, 2012).

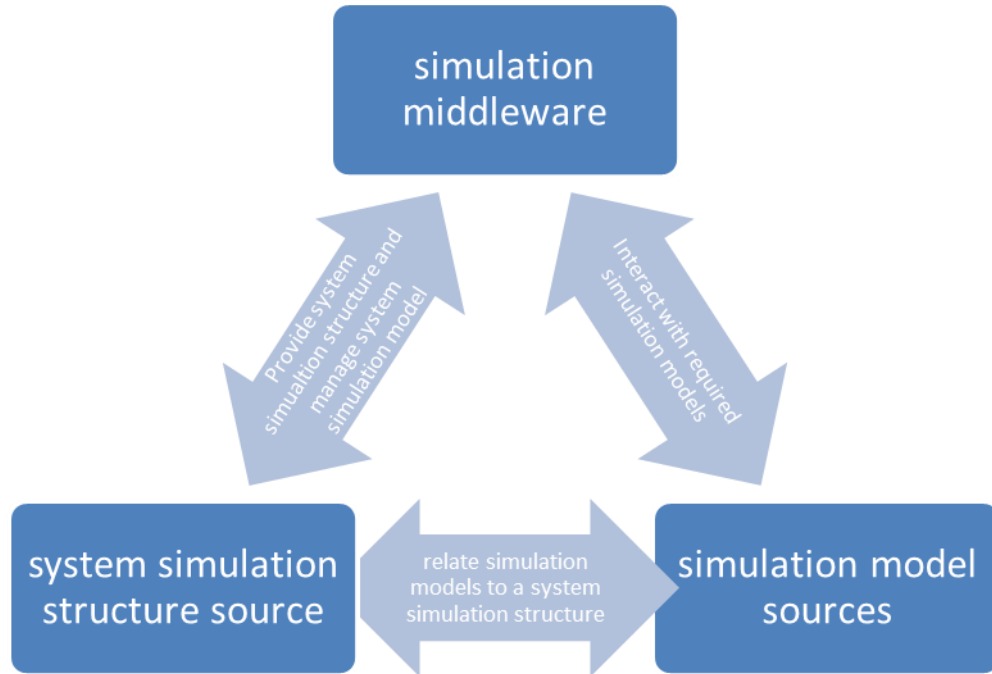
The use of simulation middleware is traditionally practiced in research projects. This practice has increased in recent years and can be seen in the publications of Audi AG and VW (Bauer, Stüber, Meller, & Gruber, 2012; Bindick, Lange, & Lund, 2012). The use of simulation middleware in Case 4 provides the possibility of linking together all the different simulation models and simulation solvers. With this technology, a co-simulation becomes achievable. The new framework for supporting and managing multi-disciplinary simulation data embedded in a PLM environment should provide the possibility to cooperate with this kind of simulation middleware. This functionality is in grey in Figure 6-26.

Instead of using simulation middleware technology, Case 3 used the software tool NX Motion and the integrated interfaces to interact with reduced flexible bodies, and Matlab/Simulink. The technology of predefined interfaces and connectors to other simulation tools is offered by many commercial simulation tools. This technology finds regular use in the industry and must be integrated into the new framework to support and manage multi-disciplinary simulation data embedded in a PLM environment as well. These simulation tools take over the solve run of an individual simulation model and the co-simulation work. So, data management has to be open to support a single simulation middleware or a simulation tool that includes the functionality of coupling simulation models.

### 6.5.2 THE ARCHITECTURAL CONCEPT APPROACH

The new framework has to interact with multiple sources. In the new framework the interaction of three main technologies can be implemented. One technology is the system-simulation structure source. Here, the simulation models are related and ordered to generate a system-simulation structure. The simulation models are embedded in a second technology, the simulation model source. The system-simulation source and the simulation model source provide the required simulation models for the simulation middleware. The simulation middleware provides the technology for a simulation integrator. For efficient simulation

integration, a specific system-simulation model is created by the simulation middleware that is managed under the system-simulation structure.



**Figure 6-25 Multi-Disciplinary Simulation Data Management Cores**

The implementation of the new framework into a PLM environment can achieve a closer cooperation between the simulation experts and other experts involved in the development process. So, the implementation of the new framework within a PLM environment provides access to a large amount of data managed in the PLM environment. TEAMCENTER provides a comprehensive framework to manage, organise and support engineering work (Siemens Product Lifecycle Management Software Inc., 2008). Therefore, TEAMCENTER assists and provides data management for engineers from the product conception through to production/customer service. System-simulation can play an important role during the product development process and should therefore be implemented into TEAMCENTER as PLM system.

With the influence of system engineering (see Section 2.7), a change to the development process was achieved. This change improves the development of mechatronic systems. With system engineering, the purpose of system requirements, system functions and the logical diagrams were increased (Department of Defense, 2001). This was summarised under RFLP, which stands for "Requirement Function Logical and Physical" (Berry, 2011). TEAMCENTER integrates RFLP by providing specific data models and structures to manage the requirements, functions, logical and physical data. For example, requirement data can be product specifications (like length or velocity), function data can be technical descriptions of

processes (like transmission of energy), logical data can be the behaviour model of a technical function solution and physical data can be the CAD model of the solution.

Access to all this data requires the integration of the new framework into an existing and given PLM system. An architectural concept of the support and management of multi-disciplinary simulation data embedded in a PLM environment can be achieved by the new framework.

## I. The High Level Architectural Concept Approach of the New Framework

In the previous sections, the requirement of a system-simulation structure was discussed. This discussion has to be implemented into the new framework. The system-simulation structure sub-orders simulation models from multiple resources (Zaeh & Baudisch, 2003). The new framework has to be positioned in such a way that the coupling and merging of all this data is possible (Hui, Liping, Li, & Tifan, 2011). Therefore, the multi-disciplinary system-simulation data management should be positioned ahead of all source data. Source data holding blocks can be behaviour models, simulation data management and simulation data stored under product structures managed with PDM. A test, such as hardware in the loop, can be a base. The source for the individual simulation models can be the product data management and system engineering data management including functions. Figure 6-26 shows how the system-simulation structure or the system-simulation data management will be positioned.

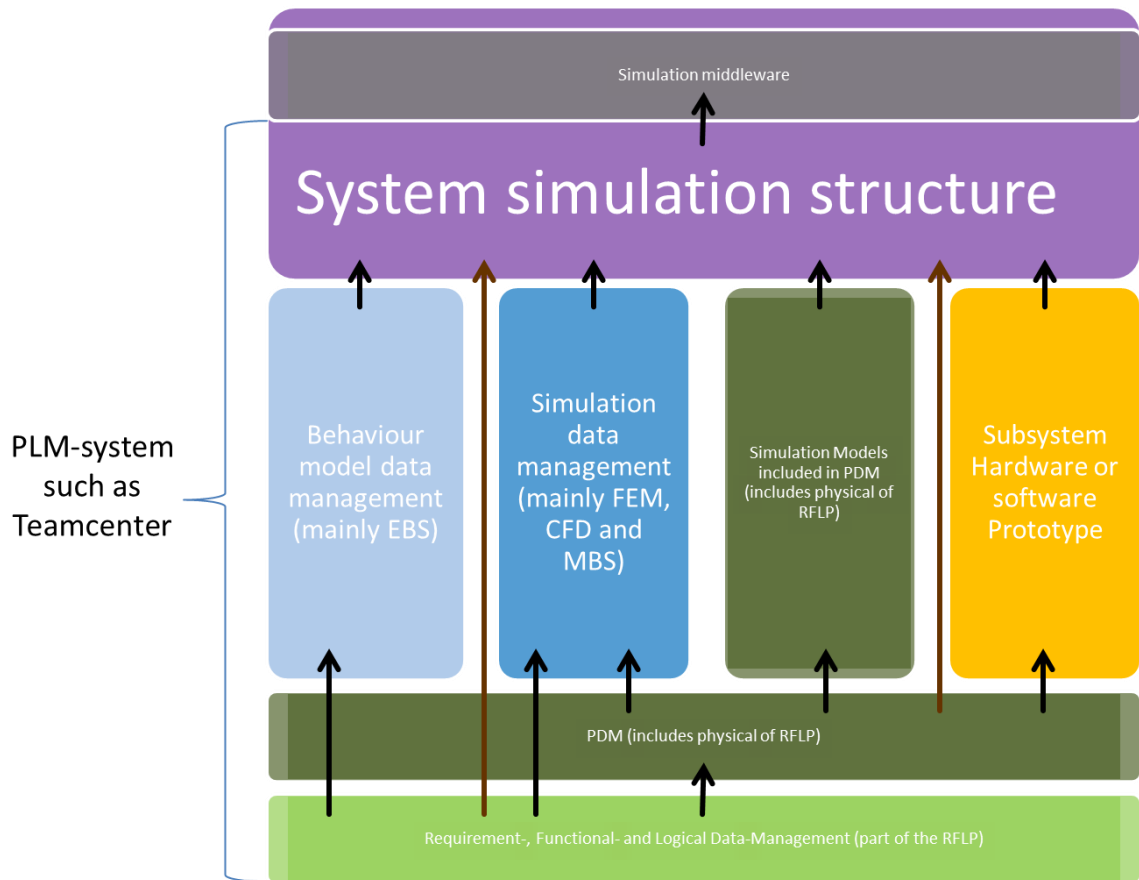


Figure 6-26 Overview of the Improved System-Simulation Architecture

Regardless of whether a simulation software tool includes different co-simulation interfaces or if the simulation middleware technology is used, the virtual models will be replaced by realistic models. These research projects are running to improve the collaboration between virtual simulations and realistic testing (Albrecht, Fremovici, Ben Gaid & Grise, 2012), this is also supported by previous study findings. An online-dependency between simulation and testing should be achieved. Actually, this technology is also aided by the LMS Company with its product family (LMS INTERNATIONAL, 2012). In order to keep a realistic prototype traceable, it has to be described and defined. An easy way to describe realistic prototypes with TEAMCENTER is to derive data structures from the engineering product structure. This functionality is in orange in Figure 6-26. However, the integration of realistic prototypes will not be pursued in the research project.

With this new architectural concept approach, following improvements can be achieved:

- The data traceability from the function or product descriptions to the system-simulation model.
- A common framework that integrates the data management of
  - function models



- product models
- CAE simulation models
- behaviour simulation models, and
- system-simulation models.
- The work on the models can be supported directly by the PLM system. Such work can be editing, viewing, versioning, revising, comparing, synchronizing, derivation etc.

In contrast to the new framework, existing and commercial frameworks for system-simulation data management such as LMS Imagine.Lab SysDM do not integrate all the data sources into one framework. Their focus is the management of behaviour models and system-simulation models that consist of sub-ordered behaviour models. Most other data is integrated by copying from other sources. This is shown in Figure 6-27.

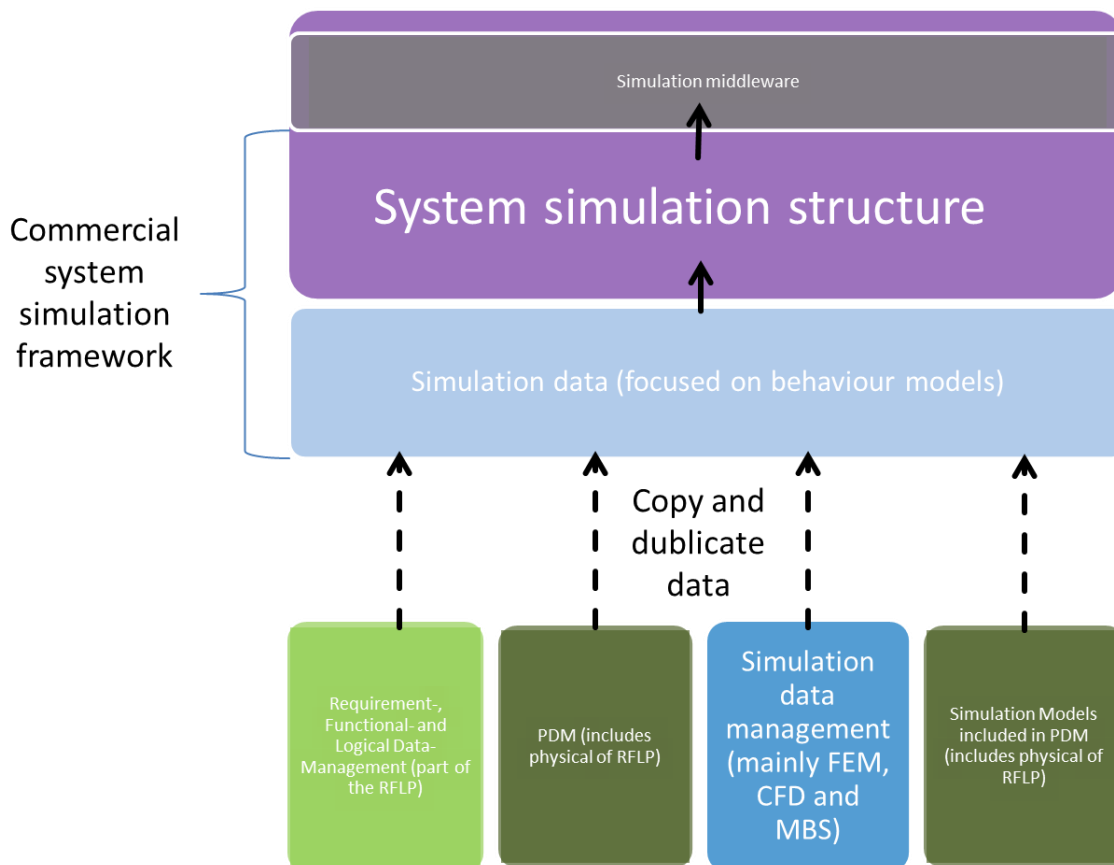
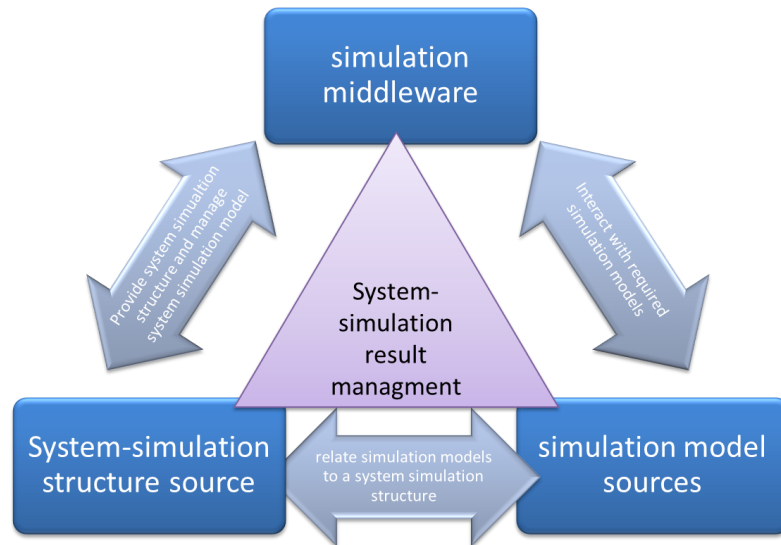


Figure 6-27 Architectural Concept of Commercial System-Simulation Data Management Applications

## II. The Advanced Architectural Concept Approach of the New Framework

For the execution of the system-simulation, the data from individual simulation models have to be supplied to the simulation middleware (McLean, Riddick, & Lee, 2005). The simulation

models should be exported from the simulation model sources. This data should be identifiable by the system-simulation structure where the simulation model files are linked. The system-simulation structure is linked with metadata technology to the simulation models. The PLM environment should be able to identify the required data and export it to an exchange folder. The simulation middleware takes the data files from the exchange directory and executes the simulation with all necessary sub-executions. The cooperation of the three cores is shown in Figure 6-25.



**Figure 6-28 Improved Four Cores Technology Architectural Concept Approach**

In the new framework, the system-simulation structure is one of four core technologies shown in Figure 6-28. The system-simulation structure has to couple and interact with the other core technologies such as simulation model sources, system-simulation result structure and system-simulation middleware. The fourth core technology is the system-simulation result structure. This interaction manages the simulation result data in relation to the system-simulation structure and to the individual simulation models. The results are generated during a system-simulation solve run. This core technology achieves:

- a unique traceability of the system-simulation results to the system-simulation and to the responsible sub-ordered simulation models.

The system-simulation result management is discussed, in detail, in Section 6.4.

Figure 6-29 shows the improved core architecture of the multi-disciplinary simulation data management by combining the four core technologies: system-simulation structure, simulation model sources, simulation middleware and system-simulation result structure. The inputs, bases and sources for the system-simulation structure are system-describing function and product structures. The inputs, bases and sources for the simulation model sources are the behaviour models, simulation data and product data management. The

results of the system-simulation are handled by a system-simulation result structure that should provide the results for the individual simulation model and system-simulation model post-process.

With this technology, huge and complex system-simulation structures can be handled, managed, supported and overviewed. Besides, the results of the system-simulations can be effectively managed with this technology.

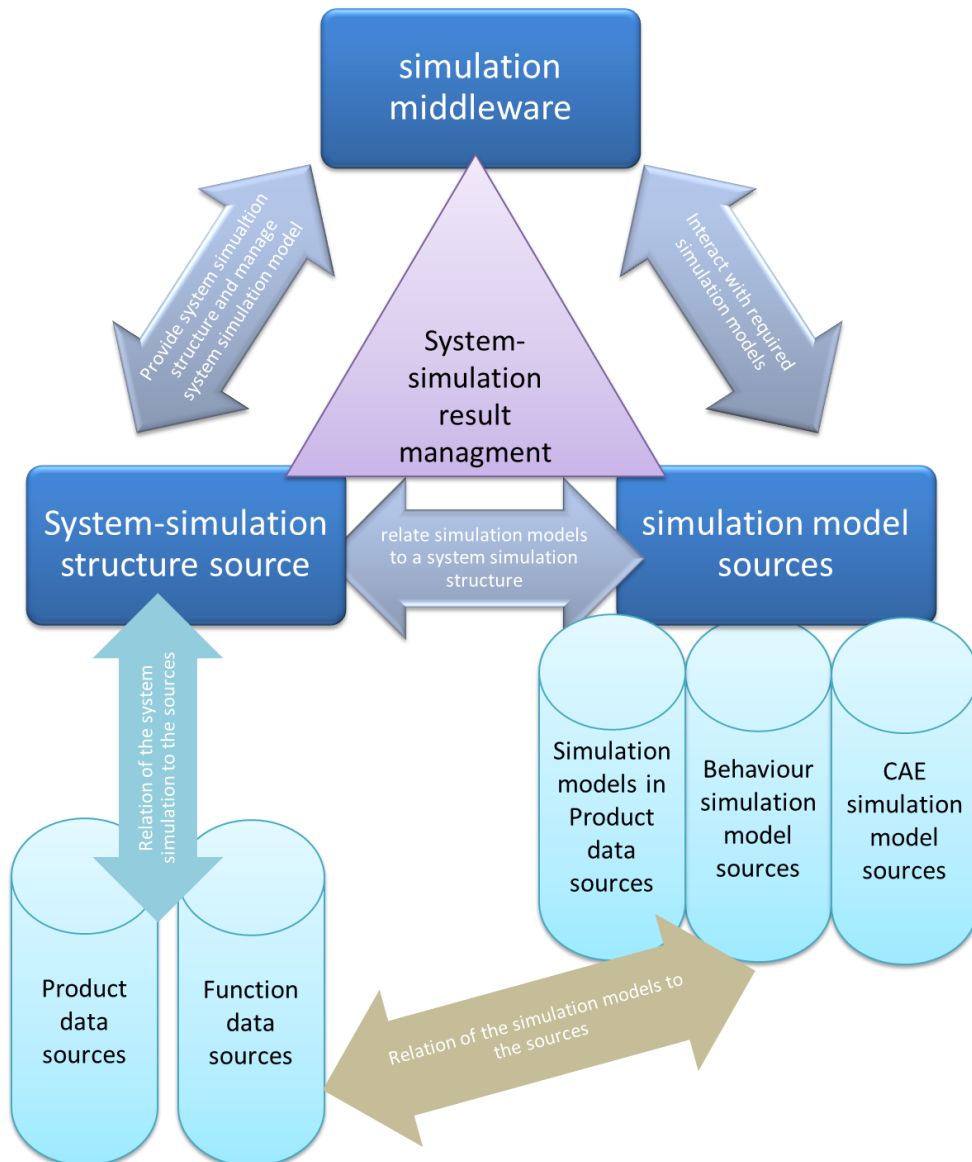


Figure 6-29 Improved Architectural Concept of Multi-Disciplinary Simulation Data Management

This new framework can be used to significantly improve engineering processes. The generation of a system-simulation requires the cooperation of multiple persons embedded in the development process. Such a process can be:

- the generation of function models and structures to represent a system. This is part of a system engineering process where multiple responsible engineers and experts are involved.
- simulation models to represent the behaviour of a function can be generated. The representation of a function can require special expertise and knowledge. Therefore, specialised experts are responsible to generate behaviour simulation models for special functions. In order to generate behaviour models for multiple functions, multiple simulation model generation specialists are required. The experts can be spread around the world and be implemented in different departments or companies (which can also be the supplier). The individual behaviour simulation models can be managed in relation to the function sources.
- the generation of a system-simulation structure. The function or product structure can help to identify the structure for the system-simulation. The required sub-simulation models can be identified based on the function structure and related simulation models.
- to generate the top simulation model. This top simulation model interacts with the sub-ordered simulation models of the system-simulation structure. Special expertise and knowledge is usually required to generate such a system-simulation model.
- to generate simulation results. These simulation results are partitioned by multiple simulation models. The new framework can be used to manage the simulation results related to the responsible simulation models and also to the system-simulation.
- to post-process simulation results. The post-processing of the simulation results often requires special expertise and knowledge. This expertise and knowledge is available from the individual simulation model and the system-simulation. The new framework can also be used to obtain the support of individual post-processing.

So the new framework can achieve:

- a common environment supporting and managing multi-disciplinary system-simulations. Therefore, the new framework can support and manage data from inception, such as the simulation source generation through to system-simulation model post-processing.

## 6.6 EVALUATION OF THE NEW FRAMEWORK

The research project worked out a new and improved framework including solutions and answers to the research questions and objectives. The technical improvements achieved by the new framework are discussed in Section 6.1 through 6.5. The main improvements are summarised to benefit-improvements in the following sections:

- Benefit of the traceability of the data.
- Benefit of the openness to support the data management of different simulation tools and simulation middleware tools.
- Benefit of the support of individual experts responsible for individual simulation models and system-simulation models
- Benefit of the support of individual experts responsible for individual simulation model post-processing and system-simulation model post-processing.
- Benefit of the synchronization of product-describing structures.
- Benefit of the derivation of system-simulation structures based on function structures or product structures.
- Benefit of the system-simulation structure as a 150% structure.

In the following sections, the improvements are discussed individually.

### 6.6.1 BENEFIT OF TRACEABILITY

In Section 6.1, it was discussed that the system-simulation structure is created by the hierarchical management of individual simulation models. Figure 6-30 shows this ordering (system-simulation structure is pale blue and the individual simulation models are dark blue.). The system-simulation structure also linked with a Sys-Sim relationship [as source or target relationship] to the source or represented function or product (see the lavender colour in Figure 6-30). This achieves a unique traceability of the simulation sources and represented functions or products.

Additionally, a system-simulation result structure is established. The system-simulation result structure hierarchically manages the individual system-simulation results. The CAE Result relationship links the simulation results by pointing to the individual simulation model or the system-simulation structure. (see Section 6.4). This is shown orange in Figure 6-30. A unique traceability of the simulation results to the simulation models is achieved.

The entry of the additional Sys-Sim, Sys-Sim Result items, item revisions and the relationships Sys-Sim target and source achieve a unique traceability of the data. The simulation model sources (behaviour model item revision, CAEAnalysis item revision or CAEModel item revision) of an individual simulation model result (managed in a Sys-Sim Result item revision) can be identified by the CAE Result relation (see the orange-coloured arrow and the dark blue box in Figure 6-30). From there, it is possible to follow the specific links of the item type down to the source of the simulation such as a product or function

(white arrow in Figure 6-30). It is also possible to identify the system-simulation result structure (Sys-Sim Result item revision) embedding the individual simulation result. Therefore, BOM view technology is used. BOM view technology helps to figure out where system-simulation results (Sys-Sim Result item revision) are embedded (the orange boxes in Figure 6-30). The system-simulation structure (Sys-Sim item revision) responsible for the system-simulation result structure (Sys-Sim Result item revision) can be identified by following the CAE Result relationship (in the pale blue box and the orange-coloured arrow in Figure 6-30). From there, it is possible to follow the Sys-Sim target or source links down to the source or represented item revision of the product or function structure (the lavender-coloured arrow to the lavender-coloured boxes in Figure 6-30).

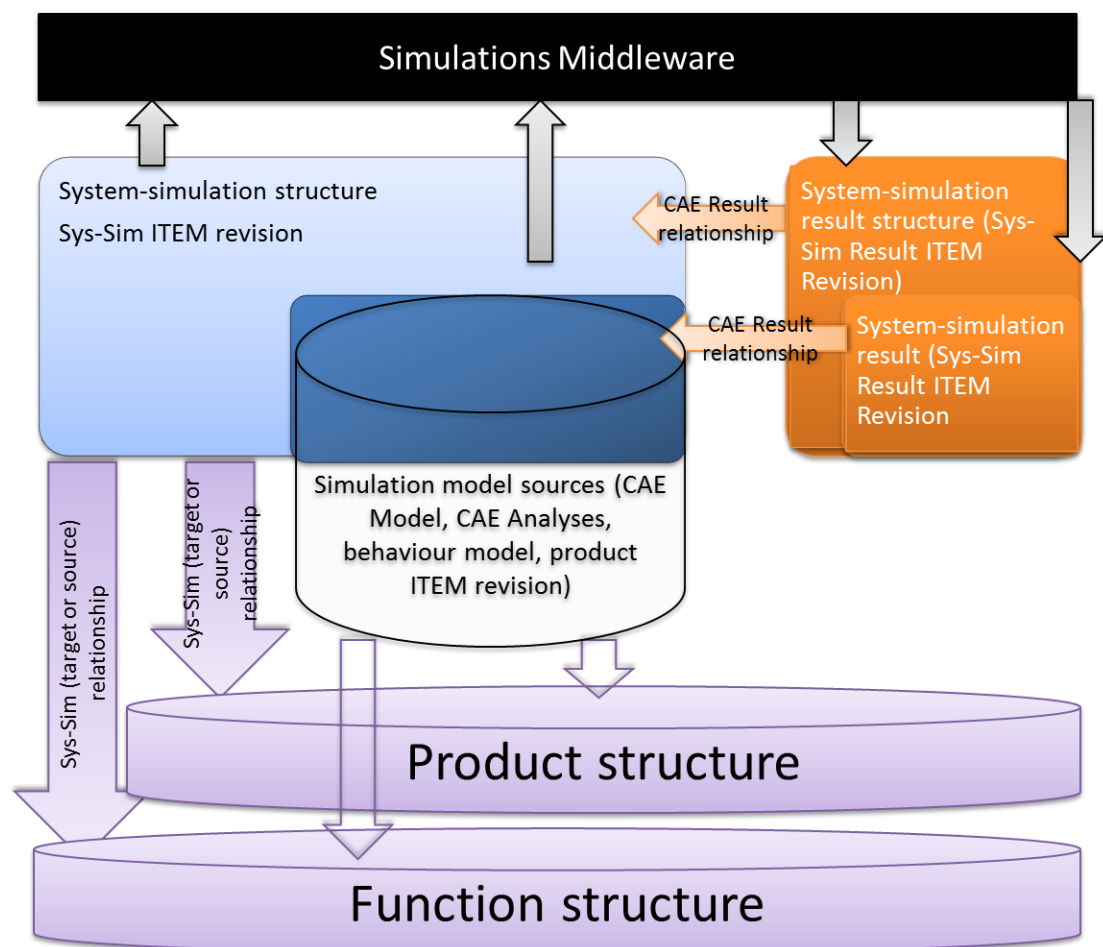


Figure 6-30 Management Architecture Achieved by the Ordering Rules

The traceability of the structure was empirically tested by theoretically transforming Cases 3 and 4 into the newly developed system-simulation structure.

The implementation of the system-simulation structure into TEAMCENTER was not technically achievable in a short space of time and hence, not achievable in my research

project time line. This means, it was not possible to test and verify the hypothesis practically. This work is documented in Sections 5.4.5 and 5.5.5. In this work no loss of traceability of the system-simulation data was identified. All data can be tracked down to their sources as well as to the structures in which they were used.

Another improvement of the new framework is its receptivity to interact with multiple simulations and simulation middleware tools, as discussed in the following Section.

### 6.6.2 *BENEFIT OF OPENNESS FOR MULTIPLE SIMULATION TOOLS AND SIMULATION MIDDLEWARE TOOLS*

The openness for multiple simulation tools means that the new framework is not restricted to cooperating only with specific simulation tools, regardless of whether it is a simulation tool for the generation of an individual simulation model or a simulation middleware tool connecting simulation models for co-simulation (see Section 6.1.2).

The ordering rules and data management are independent from simulation and simulation middleware tools. Existing technology of TEAMCENTER provides the execution of external simulation or of simulation middleware tools and includes possibilities of data export for external use. The data export is definable in detail. The execution technology for external tools and the data export is also available in TEAMCENTER for Simulation (Siemens Product Lifecycle Management Software Inc, 2012). The technology achieves the appropriation of necessary data for external simulation and simulation middleware tools and can also achieve the execution of external tools. It does not matter if the simulation middleware tool:

- is an extracted simulation middleware tool that is focused on the organization and execution of co-simulation solve runs. Such an example is ICOS from virtual vehicle in Graz, or
- is a simulation tool with focus on special simulation disciplines but includes additional functionality to cooperate with other simulation tools in a co-simulation solve run, such as NX Motion from Siemens Industry Software GmbH & CoKG.

In my SIEMENS pre-sales project “VW Systemsimulations Demonstrator”, a pilot of concept - demonstrator was generated. This demonstrator showed how the VW concept and framework of system-simulation could be supported (Bindick, Lange, & Lund, 2012). The demonstrator used the existing technologies of TEAMCENTER. In this project, the simulation middleware ICOS was implemented to TEAMCENTER and to see if an implementation of the simulation middleware TISC was possible. This demonstrator showed that the execution, supply and export of necessary data was possible for different simulation middleware authoring tools. Also, simulation tools such as OpenModelica and Matlab/Simulink and directly executable simulation models were integrated as single simulation models. The source of the model was unknown but, for example, the source could be a simulation model



written in the Modelica format and compiled to be executable (Vasaiely, 2009). The execution, supply and export of the necessary data was possible for different simulation authoring tools. So, this VW demonstrator validated the idea of interacting with a simulation integrator (middleware) and with single simulation models (see Figure 6-2).

In this project, it was not possible to use the generated ordering rules of my research project. Still, it was tested with the existing simulation structure of TEAMCENTER described in Sections 4.2 and 4.6. CAE Model item revision structure was created to represent the system-simulation structure and the CAE Analysis item revision for managing the data of the simulation middleware was used. In this SIEMENS pre-sales project, some points did not focus on system-simulation result management or the traceability of simulation models down to their sources. The interaction of the simulation middleware ICOS and the usability of other simulation middleware tools with TEAMCENTER was the focus. The requirement to support multiple simulation tools and simulation middleware tools was achievable due to the data export technology and the 3<sup>rd</sup> party software execution technology of TEAMCENTER. This existing technology is also implemented into the new framework to support and manage multi-disciplinary simulation embedded into PLM.

With the “VW Demonstrator Projekt”, it was possible to validate the openness of the new framework. Therefore, the existing technology was used in a non-standardised way; so, the demonstrator was more closed to the new framework. The existing technology in TEAMCENTER will be re-used in the new framework. The new framework improves this technology to achieve more openness and traceability. In Cases 3 and 4, the new framework was empirically validated on the reusability of different simulation models (different disciplines, tools, etc.) (see Sections 5.4.5 and 5.5.5).

The openness to support multiple simulation tools and simulation middleware tools is also important for the following improvement in supporting individual experts of individual simulation model generation and system-simulation model generation.

### 6.6.3 *BENEFIT OF SUPPORTING THE INDIVIDUAL EXPERTS OF INDIVIDUAL SIMULATION MODEL GENERATION AND SYSTEM-SIMULATION MODEL GENERATION*

A system-simulation connects different physical disciplines, mostly multiple simulation disciplines and multiple simulation tools. The individual simulation models can be generated by different experts with special expertise in the physical or simulation discipline or in the simulation authoring tool (see Section 6.5). This justifies the requirements to achieve the support of each individual simulation expert by providing a framework that supports his/her simulation model generation (pre-process) job.



Therefore, the simulation model has to be managed individually. The data of a managed simulation model has to be provided and exported to be used by an individual simulation authoring tool. An automated execution of the individual simulation authoring tool including the opening of the simulation model is useful for the experts. The achievement of the described requirement is one benefit of TEAMCENTER for Simulation (Siemens Product Lifecycle Management Software Inc, 2012). This requirement is achieved through the execution technology for external tools and the data export technology of TEAMCENTER for Simulation (see Section 6.6.2). The new framework and ordering rules re-use the individual simulation models. The individual simulation models can be generated in a managed mode with TEAMCENTER for Simulation. TEAMCENTER for Simulation provides a supporting framework for multiple and individual experts to generate and manage simulation models. This achieves the openness of the new framework to support individual experts of single simulation model generation.

The technology for external tools and the data export technology of TEAMCENTER for Simulation can be re-used in the generation of the system-simulation model. Therefore, this existing technology should be implemented into the new framework. With the integration of this technology, the supplying and export of the necessary data to run the system-simulation is achieved, as well as the execution of the system-simulation authoring tool.

An improvement will also be achieved in the case of the post-processing discussed in Section 6.6.4.

#### *6.6.4 BENEFIT OF SUPPORTING INDIVIDUAL EXPERTS OF INDIVIDUAL SIMULATION MODEL POST-PROCESSING AND SYSTEM-SIMULATION MODEL POST-PROCESSING*

With post-processing, the simulation results are analysed and interpreted. Similar to the pre-processing in the previous section, the analyst and interpreter is an expert of the simulation tool, the simulation discipline and the physical discipline. Due to this, post-processing should be carried out by multiple experts (see Section 6.5). The responsibility has to be clarified for each individual simulation model taking part in the system-simulation. Predominantly, post-processing is carried out by the same experts who worked on the pre-processing of the individual simulation models. Invariably, the simulation authoring tools, used for the generation of the simulation model, are also used for post-processing.

The new framework has to achieve the management of the result data in such a way that the result data of each individual simulation model is dependent on the simulation model. Therefore, the new framework can manage the result data of each individual simulation model in a single Sys-Sim Result item revision. The results of the individual simulation model are linked to the simulation model by the CAE Result Relationship. In this way, every result of an individual simulation model is linked to its simulation model source. The user has to identify whether the results of the simulation are created based on a single simulation run or

a system-simulation run. Managing the results by a new item type, called Sys-Sim Result item revision, achieves the ability to identify the results as system-simulation-based.

The managing of individual simulation results of a system-simulation also requires traceability to the system-simulation structure and to the other results generated by a system-simulation. Managing multiple individual simulation results in a system-simulation result structure achieves the necessary dependency. The individual simulation model results are assembled in the system-simulation result structure and, by following the structure to the next higher level, helps to identify the top system simulation result. From the higher ordered system-simulation result, a relationship to the respective system-simulation model is available. The ordering rules of the new framework provide the possibility to sub-order other system-simulation models and system-simulation results and individual simulation models, as well as individual simulation results. So, the ordering rules achieve the support of individual experts for the post-processing stage.

The new framework was tested and validated empirically in case studies 3 and 4. The used BOM view technology in TEAMCENTER is proved state of technology and used in multiple TEAMCENTER use cases. It was also used in the SIEMENS pre-sales project “VW Systemsimulations Demonstrator” mentioned in previous sections.

The ordering rules can also achieve the interaction between the system-simulation and the product and function, or their structures. This will be discussed in the next section.

#### *6.6.5 BENEFIT OF SYNCHRONIZATION WITH PRODUCT AND FUNCTION STRUCTURE*

The system can be described by the product structure but also by the functional structure. The difference between the two is that the product structure includes the description and virtual representation of the product. In contrast, the functional structure describes the functionalities of systems and subsystems. The function structure does not include how and with which parts and assemblies the functions are realised. This is carried out in the product structure. With the integration of a system-simulation structure, an additional description of the system is incorporated. The system-simulation structure can combine both, the product and the functional structure, to a virtual prototype for virtual testing (see Section 6.3).

Since the system-simulation structure is not connected to the product and to the function structure of a product, there is no possibility to synchronise the structures. Lack synchronisation of the system describing structures cause different virtual descriptions of the same system and product. Different virtual descriptions of the same system and product cause a difference of the virtual and the physical system and product and with that lack in achieving the system and product requirements. With the implementation of multiple system descriptions in the development process, the synchronization of the three structures becomes more important. The synchronization improves communication between different

departments working on different structures. To a great extent, the prime structure is the function structure describing the functions of the systems. This is an ideal starting point for the simulation and for the development departments. With the publication of a system function:

- a development expert knows what he/she should achieve with his/her development process, and
- a simulation expert can generate a first mathematical model of the function.

The lack of system function knowledge forces the development and simulation experts to search manually (by interviewing other persons) for this unmanaged knowledge, or to generate this knowledge on their own. The risk of different system function understandings is high.

With the new framework, the system-simulation structure can be synchronised with the product and function structures. This will:

- minimise the search work for information and
- reduce the risk of misunderstanding the represented function or product of the simulation.

Therefore, the new framework links the system-simulation models with the Sys-Sim source relationship to their source. The Sys-Sim source relationship appears only once in a system-simulation model and points to a product or function. This relationship makes the system-simulation model uniquely traceable to its source. Additionally, a Sys-Sim target relationship can be added to the simulation model. The relationship represents function or product represented by the system-simulation model. Using the relationship can achieve the identification of source changes. The relationships can be followed down and the revision state of the product or function can be identified. With the existing TEAMCENTER for Simulation functionalities, the item and the last state or a specific state (like the last revision with a released status) can be identified (Siemens Product Lifecycle Management Software Inc, 2012). This revision can be compared to the revisions to which the Sys-Sim target or source relationship is pointing. The comparison can be used to identify the status of the system-simulation model in contrast to the function or the product structure. Therefore, the existing functionality of TEAMCENTER for Simulation can be re-used and implemented into the new framework. The functionality has to be improved to check the system-simulation structure against the product and the function structure.

The actual state of the functionality can be seen as validated. Improvements to the existing technology could not be achieved as part of the research project. The production of a validation framework would have been too time-consuming.

The ordering rules of the new framework not only support the possibility of comparing and synchronizing the structures but also to deriving a system-simulation structure from a product structure or function structure. This will be discussed in the Section 6.6.6.

#### 6.6.6 *BENEFIT OF DERIVATION OF SYSTEM-SIMULATION STRUCTURES BASED ON FUNCTION OR PRODUCT STRUCTURES*

System-simulation structures are mostly created without predefined sources. A direct relationship to these sources might be missing because of the lack of a shared database. This can cause anomalies between the system-simulation structures and the product or functional structures. The new framework achieves:

- a shareable database storing the product, function, system-simulation structure, and
- the use of ordering rules to synchronise a system-simulation structure with a product or function structure.

This is a base to derivate the system-simulation structure from existing structures. A derivation of a system-simulation from an existing function structure or a product structure achieves a reduction of work. The manual and native creation of the system-simulation structure is time-consuming and includes a high level of information search. Additionally, the synchronization of the manually and natively created system-simulation to ongoing work on function and product structures includes high risks. An automatic and managed derivation of the system-simulation structure from a function or product structure can be based on company internal, populated regulations and can achieve similar structures. The standard based derivation of the system-simulation structure is subsequently easier to understand and similar structures can be compared more easily. Each individual function or product in a function or product structure can be checked for existing system-simulation models. In the case where a system simulation model is available, such as linked to a function or a product item revision, the automation can check whether they are useable derivation. This can be based on additional, discarded data like parameters in a PLM environment (see Section 6.3).

Such derivation of a system-simulation structure could run parts of a simulation process automatically. In Chapter 6.3.2 and Figure 6-14 the automatic run of such simulation processes was proposed. In this case, an automatic simulation process will be based on a product or function structure. Each ITEM Revision, sub-ordered to the product or function structure, will be filtered (see step 1 through 3 in Figure 6-31). Therefore filtering rules have to be defined. In this way the relevant ITEM revisions in the product or functional structure will be identified for the simulation process. For example a structure could include fastener, but such fastener should not be included in a system simulation structure. The filtering will identify and separate them. Each relevant ITEM revision will be checked after available and reusable simulation models. Therefore the relationships between source data (function or product) and simulation models and the metadata of those simulation models will be checked (see step 4 through 5 in Figure 6-31). For example an MBS Model will be available to

represent an engine. This engine will be implemented in the assembly of a motor (engine, fastener, battery, electrics, controller etc.). The MBS data will be identified by the relationship between the product or function ITEM revision and the simulation ITEM revision. Then the identified simulation ITEM revision will be sub-ordered to the system simulation structure. In some cases simulation models will be missing because reusable simulation models were not identified. In such cases placeholders for the missing simulation models could be generated and related to the represented product or function ITEM revision (see step 6 in Figure 6-31). For example a relevant electrical product or function in the motor product or function structure is identified but no reusable simulation data was related. A placeholder for simulation data could be generated and related to the electric product or function. Such placeholder will contain no actual simulation data. Nevertheless, the simulation data could be fed to the placeholder afterwards. In addition, in the case where simulation data has to be generated, the placeholder could be used to describe the environment in which the simulation data will be used.

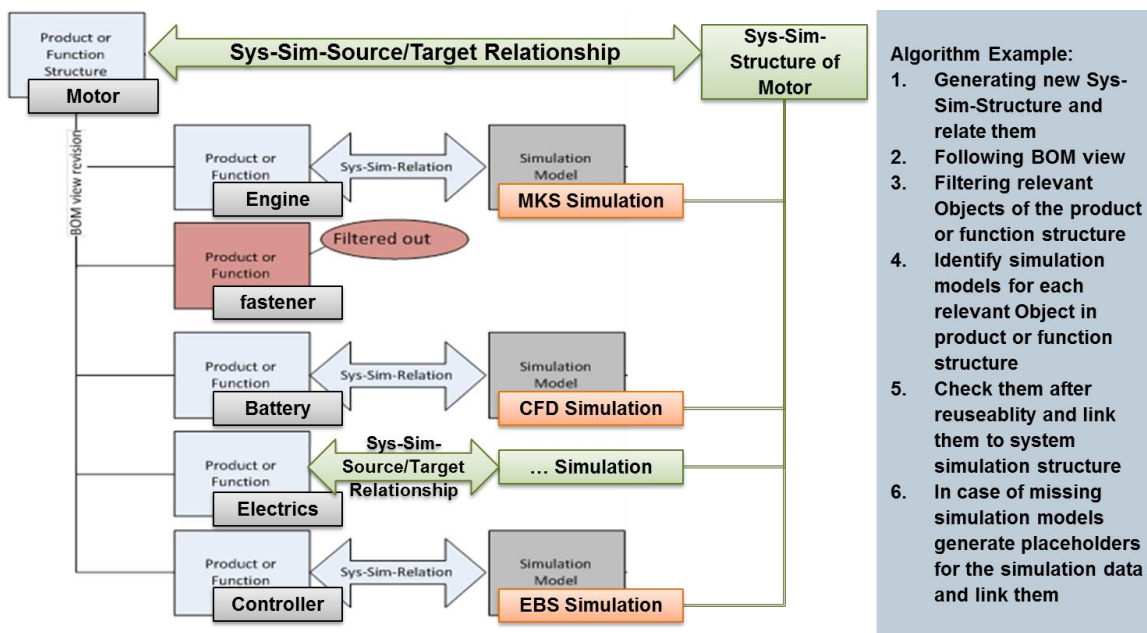


Figure 6-31 System Simulation Derivation Example

The proposed automatic generation of simulation structure and the proposed automatic simulation process will cover:

- an automatic selection, based on predefined rules and filters, of sub-ordered elements in a source structure; elements such as products or functions in a product or function structure,
- afterwards the automatic identification of simulation models related to the relevant elements of the source structure,

- and afterwards the selection of the simulation models after reusability, based on predefined rules and filters,
- then the automatic linking of the simulation models in a system-simulation structure and,
- the automatic linking, as well as generation, of the system simulation structure
- the automatic generation of placeholder to replace missing simulation data, plus the linking of the placeholder.

The ordering rules of the new structure relate the individual system-simulation models or system-simulation structures to products or product structures, or to functions and function structures. This is carried out by the Sys-Sim relationship. This relationship points from the system-simulation model to a product, product structure or to a function or function structure. The relationship is traceable from both sides and means the relationship can be used to identify a system-simulation model based on a product or product structure, or on a function or function structure. The ordering rules of the new framework provide the necessary relationships to achieve a derivation of a system-simulation structure.

In TEAMCENTER for Simulation, a technology to create simulation structures based on product structures is implemented. This technology is called structure mapping which provides the capability of defining regulations (algorithms) for the creation of simulation structures and simulation items, based on product structures. The technology runs the derivation automatically based on predefined rules. Function structures are not supported by this technology. For a re-use in the new framework, this technology has to be improved to support function structures too. With the re-use and implementation of this improved technology into the new framework, the automated and rule-based derivation of system-simulation structures can be achieved.

The new framework can also provide the ability to support variants of individual system-simulations in the same system-simulation structure. This will be discussed in Section 6.6.7.

#### *6.6.7 BENEFIT OF SYSTEM-SIMULATION STRUCTURE AS 150% STRUCTURE*

The new framework provides a system-simulation structure useable during the development process. System simulation models can be created from the conception of a product and enhanced during the development process till the end. At this point, the system-simulation model can be used for detailed system-simulations and virtual prototypes. The virtual prototype can represent different worst-case scenarios, for example, the tolerance differences of a product resulting from the production or montage. Factors like this can be respected in a virtual prototype (Mahler, 2012).



For a detailed analysis of the virtual prototype, a high level of detail is required from the simulation models. The need for detailed simulation models is dependent on both the analysis task and the assignment of the task. The following example will help to understand the need for detailed simulation models in dependency to the analysis task and the assignment of the task: The analysis task of stress due to inspired vibrations of a virtual prototype in the case of the maximum and minimum mechanical tolerances do not require a high detail simulation of the controller. On the contrary, the validation analyses task of controller routines of the virtual prototype in a working state requires a very detailed simulation model of the controller. But the mechanical simulation can be done by MBS instead FEM simulation (if no flexible body influences appears). In summary, the first analyses cases require very detailed simulation models of the mechanic. In contrast, the second analyses cases require low detailed simulation models. Both analyses cases could appear in the same development stage.

It would be ideal if both analyses cases are produced based on one system-simulation structure. Therefore, a system-simulation structure has to be able to include variants. Structures that include variants are called 150% structures because they represent more than one structure. The representation of one structure is a 100% structure. Siemens Product Lifecycle Management Software Inc. (2012) suggests that "TEAMCENTER supports two techniques of managing variants: classic variants and modular variants." Both techniques provide an opportunity to define variants of the structure. With the integration of these technologies into the implementation of the new TEAMCENTER framework, the variant generation of system-simulation structures will be achieved.

There is no use case that includes the needs to build variants of system-simulation structures. Consequently, validation of the variant generation of system-simulation structures is neither planned nor realised.

Many improvements will be achieved with the new framework to support and manage multi-disciplinary simulation. Nevertheless, some critical points are discussed in Section 6.7.

## 6.7 CRITIQUES ON THE NEW FRAMEWORK APPROACH

The new framework to support and manage multi-disciplinary simulations will be embedded into a PLM environment. With the integration of this new framework to a PLM environment a large number of improvements and benefits can be achieved. Nevertheless, the research work was not able to improve everything. Focussing on data management and process support, the following critical points can still be improved:

- The support of system-simulation interfaces in the new framework is missing.
- The support of object-oriented visualisation of the system-simulation is missing.
- The support of process-oriented simulation processes is missing.

These points are discussed in the following sections.

### 6.7.1 MISSING INTERFACE DESCRIPTIONS

Interfaces are required to link simulation tools, simulation solvers and simulation models to a co-simulation. They send the information outside a solving simulation model and receive information (McLean, Riddick, & Lee, 2005). The interface provides the possibility to connect the solving simulation model to a simulation middleware or, alternatively, to another direct solving simulation model. In this way, the interfaces are used to couple different simulation models.

Therefore, the interfaces provide individual simulation model communication ports for simulation model input and output data (McLean, Riddick, & Lee, 2005). The behaviour simulation model data management includes these communication ports (see Section 4.8 Data Acquisition of the Behaviour Models Technology) which are specified for a Matlab/Simulink data model. In the case of the new framework, the communication ports require information about:

- the kind of port (input or output),
- the type of interface (MODELISAR FMU, ICOS Interface, NX Motion Interface),
- the unit of the value,
- the limits of the value etc.

An interface type could be the MODELISAR interface (MODELISAR Consortium, September 30, 2010 & January 26, 2010). These types of communication ports are used in Case 4. The communication ports and their type should also be available for each simulation model that takes part in a system-simulation. This information should be accessible on the item revision level of the simulation model such as in the case of a behaviour model item revision.

In the system-simulation structure, the individual simulation models are connected for a later system-simulation run. This connection links a simulation model output port to a simulation model input port or to a data bus to provide data. A simulation model input port is linked to a simulation model output port or to the data bus to receive data. The behaviour simulation model data management includes such connectors (see Section 4.8). The connectors (including the output port, input port or data bus) should also be available for each system-simulation structure. This information should be accessible on the Sys-Sim item revision level of the simulation model such as in the case of a behaviour model item revision.

The connections should be supported by object-oriented visualisations. This will be discussed in Section 6.7.2.



### 6.7.2 MISSING OBJECT-ORIENTED VISUALISATIONS OF THE SYSTEM-SIMULATION

The structure of a system-simulation can be complex. This can be seen in Cases 3 and 4. In order to keep the system-simulation both workable and simple to review, object-oriented visualisations could be incorporated. Object-oriented visualisation means the virtual visualisation of the system-simulation architecture. An example could be a visual box that represents a Sys-Sim item revision or a line or arrow that represents a connection between two communication ports. The object-oriented visualisation should monitor:

- the structure of the system-simulations
- the individual simulation models
- the input and output ports of the individual simulation models
- if available, the data bus, and
- the connections between the output ports, input ports or data bus including the data flow.

The user may require specific information such as:

- the kind of port (input or output)
- the type of interface (MODELISAR FMU, ICOS Interface, NX Motion Interface)
- the unit of the value
- the limits of the value etc.

Also, dependent data may be required such as sources of a simulation, or a represented product or function of a simulation model.

Object-oriented visualisation increases the usability and attractiveness of the new framework to support and manage multi-disciplinary simulation embedded in PLM. It helps to include the support of process-oriented simulation processes into the new framework and into object visualisation. This will be discussed in Section 6.7.3.

### 6.7.3 MISSING FRAMEWORK FOR PROCESS-ORIENTED SIMULATION PROCESSES

The focus of the new framework to support and manage multi-disciplinary simulations embedded in PLM is on system-simulation. The process-oriented simulation process was a discursive exercise (see Section 6.2). This discussion could lead to an improvement in existing technology. However, the integration and combination of system-simulation and process-oriented simulation processes is not detailed enough to be implemented.

It should be stated that a more common and open kind of support and management of multi-disciplinary simulation embedded in PLM will require both the support and management of

system-simulation and process-oriented simulation processes. This could be achieved by expanding the new framework. Therefore, detailing and solution analysis of supporting and managing process-oriented simulation processes embedded in PLM is required. A more detailed discussion on process-oriented simulation processes, the perceived need for it as well as possible achievements is available in Section 6.2.

## 7 CONCLUSION AND FURTHER WORK

### 7.1 THE RESEARCH OBJECTIVES

The overall aim of this research was to research, propose, design and develop a new framework that supports and manages multi-disciplinary and inter-disciplinary system-simulation embedded in a PLM environment. The main functionalities and metadata structures of the new framework have been identified and optimised. Such functionalities and metadata structures will involve a new method for ordering and relating data in a multi- and inter-disciplinary simulation context (see section 7.2). The multi-disciplinary simulation data and their collection processes, the existing PLM software and their applications have been analysed. In addition, the inter-disciplinary collaboration of various simulation software applications has been analysed and evaluated. The new framework integrates the identified and optimised functional and metadata structures to support and manage multi- and inter-disciplinary simulation in PLM system environment. To achieve the overall aim there are the three planned research objectives as discussed in Chapter 1:

- (1) The first objective of this research is to analyse and evaluate multi-disciplinary simulation data and their collection processes, the existing PLM software and their applications to identify and optimise the required functional and metadata structures to support and manage multi-disciplinary simulation data and processes embedded in a PLM environment. To achieve this objective, data holding and data managing software solutions are evaluated and the best software solution (the best in relation to the research objectives) was selected. In this PhD project, TEAMCENTER was used as the case study. The restriction of the research project to a reduced number of software applications was necessary to keep the research project on time and within budget. The data acquisition details the functions and metadata structures of this data holding and data managing software solution. In addition, case studies with multi-disciplinary simulations are sampled, discussed and evaluated from the viewpoint of data management, data dependencies, simulation processes and collection processes. The analyses of the case studies (including the multi-disciplinary simulations and the implementation to TEAMCENTER) are used to identify and optimise functional and metadata structures to support and manage multi-disciplinary simulation data and processes in a PLM environment.
- (2) The second objective is to analyse and evaluate inter-disciplinary collaboration of various simulation software to identify and optimise the required functional and metadata structures to support and manage inter-disciplinary simulation data and processes embedded in a PLM environment. Multi-disciplinary case studies are sampled, discussed and evaluated from the viewpoint of inter-disciplinary collaboration, simulation software, simulation processes and data-dependencies. As discussed above, the cases are implemented in the PLM environment TEAMCENTER.

The analyses of the cases are used to identify and optimise the required functional and metadata structures to support and manage inter-disciplinary simulation data and processes embedded in a PLM environment.

- (3) The third and last objective is to research, propose, design and develop a new framework that integrates the identified and optimised functional and metadata structures to support and manage multi- and inter-disciplinary simulation in a PLM system environment. This objective can be achieved based on the results of the first and the second objectives identifying and optimizing the required functional and metadata structures to support and manage multi- and inter-disciplinary simulation data and processes embedded in a PLM environment. In summary, the third objective of the research is to implement the identified and optimised functional and metadata structures into a new framework. Nevertheless, the identified, optimised functional and metadata structures imply a new method of ordering and relating data in a multi- and inter-disciplinary simulation context (see section 7.2). To do this, five PLM environments have been evaluated and the best one (in relation to the research project) has been identified. The restriction of the ongoing research to this PLM environment was necessary to keep the research project on time and within budget. Furthermore, existing functions and metadata structures of the identified, best PLM system have been acquired. The functions and metadata structures of the identified PLM system have been analysed and optimised. The optimised functions and data structures have successfully been integrated into the new conceptual framework that supports and manages multi-disciplinary and inter-disciplinary system-simulations embedded in a PLM environment. To validate the new framework, two cases have been empirically implemented in the new framework with existing functions.

The findings and achievements of this project are contributions to both new knowledge generation in multi-disciplinary and inter-disciplinary system simulation and the practice, as discussed in Sections 7.2 and 7.2, respectively.

## 7.2 CONTRIBUTIONS TO THE NEW KNOWLEDGE GENERATION

In Chapter 6 a new framework for improving the support and management of multi- and inter-disciplinary simulation data in a PLM environment was proposed. Thereby the new framework will use new forms of data ordering and data relationships and, based on those orderings and relationships, provide derivation and synchronisation bases. This could therefore be regarded as a new method. The new method will provide enhanced and new knowledge about ordering and relating rules to the support and management of multi- and inter-disciplinary data. Thereby the new method has achieved a level of independence. Additionally, the implementation of the new method will not be limited. To keep the research

project on time and within budget it was necessary to reduce the number of data management tools (TEAMCENTER) as well as data authoring tools. So, the research project explored the new method and conceptually implemented it into the new framework based on TEAMCENTER (see Section 7.3).

In summary, it is believed that this project has made the six main contributions to new knowledge generation, as follows:

- (1) *The first contribution is the New Method and New conceptual Framework to Enhance the Support and Management of Multi-Disciplinary System-Simulation.* The research project has proposed and developed the new method and new conceptual framework to enhance the support and management of multi-disciplinary and inter-disciplinary system-simulation. So, the new method contains the ordering and relating rules of data used to improve the traceability of the data. These ordering and relating rules will describe the management of data such as data-containers (the different types of ITEM Revisions in the new Framework) and the relations (the CAE relations and BOM views). The new method will improve the traceability and interaction of the data in a system-simulation context. The implementation of the new method will not be limited but is focused in the research project on the new framework (such as discussed in the introductory section). The new framework includes the new and enhanced orderings and relationships of data that have to be recognised to ensure a more universal, multi-disciplinary and inter-disciplinary support and management of system-simulation data, the individual simulation models and simulation results. In addition, the new concept includes an approach to enhancing PLM systems for supporting and managing multi-disciplinary system-simulation.
- (2) *The second contribution is the New System-Simulation Oriented and Process Oriented Data Handling Approach.* This contribution differentiates between system-simulation oriented and process oriented simulation data management. Both categories are using multi-disciplinary simulation data. But these are obtained through different kinds of processes (parallel and serial processes). The thesis presents two different data management approaches for the two different kinds of processes and makes them distinguishable from each other. In addition, the thesis presents new knowledge about the data management of both simulation categories and the interaction between both kinds of simulation processes. The PhD thesis discusses the current inability to distinguish between the two approaches through current methodology and commercial software applications. A conceptual solution is proposed by the thesis.
- (3) *The third contribution is the Enhanced Traceability of System-Simulation to Sources and Represented Products and Functions.* It enhances the system-simulation-structure and the traceability to sub-ordered individual simulation data. So, knowledge is enriched regarding the management of different simulation data and the relationships between

them. In addition, knowledge is also enriched regarding the traceability between simulation data, source data and represented data. The new method is enhanced to support individual simulation model generation (and generation processes) and the sampling of the individual simulation models to a system-simulation structure. Thereby each individual simulation model could have its own source and targets and the system simulation structure could also have its own source and targets. The new method is independent of software applications such as source authoring-software, individual simulation model authoring-software, inter-disciplinary simulation software and data holding software. So the implementation of the new method will not be restricted but focused in the research project on the new framework (such as discussed in the introductory chapter).

- (4) *The fourth contribution is the New System-Simulation Derivation Approach.* The thesis discusses a system-simulation-structure derivation approach. The new approach is founded on enhanced knowledge about system-simulation derivation, search algorithms, data traceability and derivation rules. The derivation approach is based on the data ordering and tracing rules of the new method. The derivation approach expands the new method by passing comprehensive and traceable data. The passing identifies relevant products or functions as well as relevant and dependent simulation data. The identified simulation data will be sampled under a system-simulation structure. Pre-definable rules and algorithms will help to run these processes automatically.
- (5) *The fifth contribution is the New Approach for the Synchronisation of System Describing Structures:* The thesis presents a new approach to the synchronisation of system describing structures. The new approach is based on enhanced knowledge about system-describing data, data relations and data traceability, synchronisation algorithms and rules for the synchronisation. The synchronisation approach is based on the data ordering and tracing rules of the new method. The synchronisation approach expands the new method by passing comprehensive and traceable data. The data dependencies of the starting structure and sub-ordered elements will be compared to source and/or target of the starting structure/ structures and sub-ordered elements. In this way, anomalies between compared structures will be identified. The anomalies should be illustrated in order to resolve them. The comparison as well as the solving of anomalies could be automated using rules and algorithms.
- (6) *The sixth contribution is the Enhanced System-Simulation Result Data Handling Approach.* Another major outcome of the research project is the enhanced knowledge about simulation result data management, individual simulation result management and the traceability between result data and simulation data, the related individual simulation models and the system-simulation structure. The system-simulation result data handling expands the new method by adding ordering and relationship rules to the result data. These rules achieve a result data handling that supports the post-

processing of the individual simulation as well of the multi- and inter-disciplinary simulation (system-simulation). The implementation of the new method will not be restricted but focused in the research project on the new framework (such as discussed in the introductory chapter).

The research project was focused on multi- and inter-disciplinary system simulation in a product development context. This puts huge constraints on the research project. Nevertheless, these constraints as well as the restrictions of applied software applications were required to keep the research project on time and within budget. So, even though these restrictions had an influence on the research project, they did not hamper the research results. The new method to Enhance the Support and Management of Multi-Disciplinary System-Simulation will be applicable in multiple and different kinds of system disciplines. Ideas exist to map the new method to simulate financial, business or ecological systems. For example the new concept could be used in financial simulations such as the flow of money between banks. In this case, it could be used for bank stress tests simulations such as those implemented after the last financial crisis. The new method could also be applied to simulate the efficiency of investments and budget distribution throughout a company. Or, the new method could be used to simulate the human influence on the environment to improve the understanding of the earth's ecology system.

Nevertheless, the new method will be implementable in many other systems and contexts. Furthermore, the new method will be a new method to generate more complex and more common virtual prototypes of systems closer to the reality. However, additional research will be required to map the method. So the new knowledge about this new method and the above five points will provide a solid basis for ongoing research and implementation.

From a historical point of view behaviour models have been seen to be used for mechatronic or system simulation (see the sections 4.8.1 4.8.2). The behaviour models were seen to represent the behaviour of functions. EBS and equation based simulation authoring tools will provide good technologies to model the behaviour of a function. Nevertheless, other kinds of simulation techniques, such as FEM, MBS, CFD and others could also be used to simulate behaviour. But these simulation techniques are not popular for behaviour model generation as they produce numerous results including behaviour representations. So the behaviour results have to be identified and extracted from the countless results available. Additionally, pre-processing, solving and post-processing will require more time and investment, leading to reduced flexibility for modification when compared to the EBS technique. Reduced interaction between these simulation techniques plus the reduced flexibility will make them unpopular for use in a system simulation context. But, in contrast to the EBS technique they will provide greater accuracy and detailed results meaning they will be popular in applications where such accuracy and detail is required. Nevertheless, the new framework will combine the EBS model approach, with the FEM, MBS, CFD, etc. simulation model approach. Furthermore, the new framework will provide increased support, management and flexibility for the interaction between these simulation approaches. So the focus of EBS



for behaviour model generation could be expanded to other simulation techniques as well. Lastly, additional investments in the FEM, MBS, CFD, etc. simulation model approach into simulation model reduction or simulation model performance improvement would be advantageous. One approach will be discussed in (Dr. Hartmann & Mahler, 2013) attached to section 9.4.

In the next section the contribution to the practice and the result by the implementation of the new method into the new framework will be discussed.

### 7.3 CONTRIBUTION TO THE PRACTICE

The isolated subsystem-simulation has become a thing of the past; the current trend is to simulate increasingly complex physical systems and products as a composition of sub-systems from multiple domains (Engelson, 2000). In order to build a multi-disciplinary modelling and simulation environment, it is necessary to adopt a development approach which moves from a single disciplinary simulation model to a comprehensive disciplinary simulation model of different disciplines (Ai, Chen, Wan, & Xiong, 2011). This can be achieved with the new framework for supporting and managing multi-disciplinary system-simulation in a PLM environment. So, the new framework can be used to reduce costs and investments involved in engineering product design and development. In particular, the potential values (see also the benefits in Section 6.6) of the new framework are:

- (1) Reduction of costs, resources and working time in the generation of individual simulation models through the more effective re-use of individual simulation models in system-simulation context. All of the simulation models are provided by the new framework. The identification of simulation models are supported by automatic search, derivation and relation technologies.
- (2) Reduction of costs, resources and working time in the generation of the individual simulation models through the effective pre-defining and preparing of individual simulation models. So, placeholder / data holder objects are useful when combined with automatable derivation and relation technologies. Such placeholder / data holder objects can be used to represent the required simulation model data, and can be related (automatically during derivation processes) to the input sources of the simulation. So the simulation model can be generated after the generation of the metadata. That means that the relations to the input source data will be present to provide a foundation for the simulation model generation.
- (3) Reduction of costs, resources and working time in the system-simulation review and authoring software through the easily reviewable system-simulation-structures with relations to the input-sources, such as products and / or functions. So, the system simulation structure adopts intelligent relation technologies. These kinds of

technologies relate to the embedded individual simulation models, the system simulation input sources, the represented products and / or functions and the system simulation results.

- (4) Reduction of costs, resources and working time in the re-editing of data through the easy and 'authoring-software independent' update of system-simulation-structures, product-structures and function-structures. So, the new framework provides a synchronisation and comparison technology between the related simulation, product and functional structures.
- (5) Reduction of costs, resources and working time in post-processing through the effective, distributed and cohesive post-processing and interpreting of system-simulation-results. So, the individual simulation results and the system simulation results could be post-processed individually even though they are linked to each other.
- (6) Reduction of costs, resources and working time in documentation and data security through effective, easy and unique traceability of data. So, each simulation model is related to the source and represented data. In addition, there are relationships between system simulation data, individual simulation data and system-simulation results data.
- (7) Reduction of costs, resources and working time in workarounds and customisations through an increased openness and flexibility in simulation software interactions with the data holding system. The existing technologies of TEAMCENTER are implemented into the new framework. In addition, the technologies and architecture are enhanced for system simulation data support and management.

Although the proposed and developed conceptual framework has not been implemented (that requires a considerable amount of resources), it can be expected that the above seven values and benefits discussed in Section 6.6 will lead to significant advances in the simulation of new product design and development over the entire lifecycle, bringing substantial practical value to the manufacturing sector.

## 7.4 FURTHER RESEARCH

Due to time and other resource limitations as well as the huge undertaking required in implementing the proposed new framework, it is evident there are some limitations to this PhD thesis. Five areas have been identified where further work needs to be carried out in order to improve the quality of this project:

- (1) *The expanded industrial sector and product design, and development processes.* The new framework was proposed and developed based on the case studies. Due to time and resource limitations, the case studies have been focused on limited industry sectors (mostly automotive) and their product development processes. Additional enhancements to knowledge could be achieved through expanded disciplines and orientations of the case studies. Industrial expansions could be applied to other manufacturing sectors, and as such, system engineering and product development process expansion could include chemical, biological and medical processes etc.
- (2) *Parameter oriented system and production description in the new framework.* The new framework achieved the support and data management of system-simulation during parameter studies. However, the parameter study approach has not been systematically applied to the case studies. Nonetheless, parameter studies play an important role in system development processes and could be used to optimise systems or products. Further work should be carried out to investigate which parameters need to be used to represent the systems or products and how to integrate these parameters into the new framework.
- (3) *The improved user interface design of the new framework.* The application of a new framework may be complicated. The new framework supports and manages data. But, the ordering and relating of data tends to be complex. So, from a practical point of view, users have to understand the framework theory, leading to easy and intuitive interactions between user and this new framework. Therefore, the theoretical complexity of the framework has to be addressed, especially in the area of user interfaces.
- (4) *The automation of simulation generation processes.* The enhancements in the area of support and management of simulation generation processes will reduce effort in the generation of high performance processes. Besides, the industry will be interested in non-expert-simulation-generation-processes to achieve a considerable number of validation and verification simulations. Large amounts of simulation data will provide more data for large amounts of system-simulations. Automation of simulation processes could enhance the simulation generation process. So, the automatic generation of simulation models should be implemented in such a way to provide a practical solution without the user having specialist simulation knowledge. As a result, the new framework could be used by non-experts.
- (5) *Enhancement of the individual simulation models.* There will be a bottleneck in the performance of the system-simulation. The performance of individual simulation models through a solve run reduces the performance of the system-simulation solve run. So, a bad performance of the individual simulation model will lead to a bad performance of the system-simulation. In addition, suppliers of individual simulation models have a vested interest in keeping the context of the simulation models secret. Nevertheless, individual simulation models should be usable in a system-simulation

context through a common and universal interactive technology. Besides, the system-simulation requires specific interfacing technologies to interact with the individual simulation models. In summary, it will be necessary to enhance individual simulation models to provide a high performance solve run keeping the content secret yet interfacing through a universal approach.

## 8 REFERENCES

- AberdeenGroup. (January, 2009). *How SMBs Save Time and Mmoney by Reducing Physical Prototypes with Simulation*. www.aberdeen.com: AberdeenGroup - Homepage.
- Abramovici, M., & Schulte, S. (2005). Wege aus der Strategiekrise in der Automobilindustrie (The way out of the strategy crises of the automotive industry). *Data-Management-Magazin*(1),  
[http://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CC4QFjAA&url=http%3A%2F%2Fwww.itm.ruhr-uni-bochum.de%2Findex.php%2Fcomponent%2Fdocman%2Fdoc\\_download%2F65-plm-wege-aus-der-strategiekrise-in-der-automobilindustrie&ei=EjtiU8](http://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CC4QFjAA&url=http%3A%2F%2Fwww.itm.ruhr-uni-bochum.de%2Findex.php%2Fcomponent%2Fdocman%2Fdoc_download%2F65-plm-wege-aus-der-strategiekrise-in-der-automobilindustrie&ei=EjtiU8).
- Abramovici, M., & Schulte, S. (2006). PLM - Stand der Technik und Entwicklungsperspektiven (PLM - stage of the technology and development-perspectives). "*Robuste, verkürzte Prozessketten für flächige Leichtbauteile*" Tagungsband zum Industriekolloquium 2006 des SFB 396. Bamberg: Meisenbach.
- Ai, H., Chen, L., Wan, L., & Xiong, T. (2011, June). A collaborative platform for complex product development based on multi-domain unified modeling and simulation. *Wuhan University journal of Natural Sciences*, 16(3), 206-212.
- Albers, I. I.-I. (2011). *PDM-System Product-Data-Management*. Retrieved 08 2011, 22, from <http://www.ipek.kit.edu/202.php>
- Albrecht, A., Fremovici, M., Ben Gaid, M., & Grise, C. (2012). Multi-Model Integration from MiL to HiL: A New Simulation Approach to address the Modern Powertrain Development Challenges. *Virtual Powertrain Conference*. Sao Paulo.
- Altair Hyperworks. (2012, 04 30). *Altair Hyperworks HomePage*. Retrieved 04 30, 2012, from [http://www.altairhyperworks.com/pdfs/product\\_brochures/HW\\_ADM\\_Web.pdf](http://www.altairhyperworks.com/pdfs/product_brochures/HW_ADM_Web.pdf)
- ANSYS. (2011, October 19). *Ansys*. Retrieved 04 08, 2012, from Ansys:  
[http://www.ansys.com/staticassets/ANSYS%20UK/staticassets/UGM\\_Downloads/Mike%20Slack%20-EKM%20Presentation.pdf](http://www.ansys.com/staticassets/ANSYS%20UK/staticassets/UGM_Downloads/Mike%20Slack%20-EKM%20Presentation.pdf)
- ANSYS. (2012, 04 16). *IDAC Integration Design & Analysis Consultants Ltd*. Retrieved 04 16, 2012, from <http://www.idac.co.uk/products/downloads/EKM/ekm-overview.pdf>
- Ansys, INc. (2011, 04 26). *IDAC Integration Design & Analysis Consultants Ltd*. Retrieved 04 26, 2012, from <http://www.idac.co.uk/products/downloads/ekm.pdf>
- Arnold, M., & Schierz, T. (2009). Effizienz und Robustheit numerischer Kopplungsalgorithmen im MODELISAR Co-Simulation-Interface. *Tagungsband ASIM-Treffen STS/GMMS 2009*

- und DASS 2009 (pp. 51-57). Dresden: Fraunhofer-Institut für Integrierte Schaltungen IIS, Institutsteil.
- Bauer, S., Stüber, N., Meller, F., & Gruber, K. (2012). Entwicklung einer Co-Simulationsprozesskette und deren Integration in den Prozess (development of a co-simulation-process-workflow and the integration into process). *SIMVEC Berechnung, Simulation und Erprobung im Fahrzeugbau 2012, 16. Konferenz* (pp. 609-616). Baden-Baden: VDI Verlag GmbH; Düsseldorf.
- Benbasat, G. I., Goldstein, D. K., & Mead, M. (1987). The Case Research Strategy in Studies of Information Systems. *Management Information Systems Quarterly*, 11(3), 369-389.
- Benedikt, M., Zehetner, J., Watzenig, D., & Bernasch, J. (2012, July). Moderne Kopplungsmethoden - Ist co-Simulation beherrschbar? (modern coupling methods - Will cosimulation be controllable?). *NAFEMS Magazin 2/2012(22)*, pp. 63-75.
- Berry, P. W. (2011). *System Engineering; Three-day Introduction 20-22 Juni 2011*. Köln: Siemens PLM.
- Bharadwaj, A. S. (1998). Multi-Domain Modeling and Simulation of Mechatronic Systems. *Actuators Automotive Electronics Series PT-74 Progress in Technology*, 74, 369-374.
- Bindick, D., Lange, C., & Lund, D. (2012). Eine virtuelle Entwicklungsplattform zur gekoppelten Simulation am Beispiel Gesamtfahrzeug (A virtual development framework for coupled simulation to use for vehicles). *NAFEMS deutschsprachige Konferenz* (pp. 357-360). Bamberg: NAFEMS.
- Blochwitz, T., & Otter, M. (2011, March). The Functional Mockup Interface for Tool independent Exchange of Simulation Models. *Modelica'2011 Conference*, (pp. 20-22).
- Boberg, F. (2008). Examensarbete Institutionen för systemteknik Department of Electrical Engineering Linköping 2008. *Simulation of dynamic and static behavior of an electrically powered lift gate*. Linköping, Sweden: Linköpings universitet.
- Boy, J., Grau, M., & Trautmann, T. (2010, März). Management und interdisziplinäre Integration von CAE Daten. *NAFEMS Magazin (1/2010)*, 1(15), pp. 46-58.
- Brendel, R., & Kühner, S. (2004, Okt). Automatisierte Routine ist das Ziel (Automised routines are the goal). *Der Konstrukteur*, 10, p. 62.
- Brown, B., & IBM Corporation. (2011, December). Model-based systems engineering: Revolutionar evolution? *IBM Software Thought Leadership White Paper*. IBM.
- Chrisofakis, E., Junghanns, A., Kehrer, C., & Rink, A. (2011). Simulation-based development of automotive control software with Modelica. *8th International Modelica Conference 2011* (pp. 20-22). Dresden: Modelica Association.

- CIMdata Inc. (2011). *Simulation Lifecycle Management "More than data management for simulation"*. 39009 Research Park Drive, Ann Arbor, Michigan 48108, USA: CIMdata, Inc.
- Clusterland Oberösterreich GmbH. (2012, 05 12). *Erhöhung der Standzeit und Reduktion der Taktzeiten*. Retrieved 05 12, 2012, from Kunststoff KC Cluster: [http://www.kunststoff-cluster.at/files/Projektblatt\\_Antihaftbeschichtung.pdf](http://www.kunststoff-cluster.at/files/Projektblatt_Antihaftbeschichtung.pdf)
- Commerell, W., Mammen, H.-T., Panreck, K., & Haase, J. (Oktober 2008 ). Simulation technischer Systeme Anforderungen und Perspektiven (simulation of technic system-requirements and perspectives). In R. Markus (Ed.), *13. ASIM Fachtagung Simulation in Produktion und Logistik Berlin 1./2. Oktober 2008* (pp. 1-10). Stuttgart: Fraunhofer IRB Verlag.
- Consortium, M. (2011, August 18). *MODELISAR*. Retrieved August 18, 2011, from MODELISAR - TOOLS: <http://modelisar.org/tools.html>
- consortium, M. (October 12, 2010). *Functional Mock-up Interface for Co-Simulation*. ITEA2 Information Tecnology for European Advancement. MODELISAR consortium.
- Cummings, T. (2006, July). *Why is EDM/PDM relevant?* Retrieved from HighBeam Research: <http://www.highbeam.com/doc/1G1-14936011.html>
- dajo Autoteile. (2013, 07 23). *dajo Autoteile*. Retrieved from [http://www.dajo-autoteile.de/uploadbilder/9752\\_IMG\\_6181\\_1.jpg](http://www.dajo-autoteile.de/uploadbilder/9752_IMG_6181_1.jpg)
- Dassault Systems. (2010, 03 23). *CD-Adapco*. Retrieved 04 19, 2012, from CD-adapco European Conference: <http://www.cd-adapco.com/pdfs/presentations/eu10/SIMULIA.pdf>
- Dassault Systems. (2011, Nov). *Dassault Systems*. (D. Systems, Ed.) Retrieved 04 22, 2012, from Simulation Lifecycle Management; Manage & Secure Simulation Intellectual Property: <http://www.3ds.com/fileadmin/PRODUCTS/SIMULIA/PDF/brochures/SIMULIA-Simulation-Lifecycle-Management.pdf>
- Dassault Systems. (2011, 11). *Dassault Systems*. (D. Systems, Ed.) Retrieved 04 22, 2012, from Simulia Execution Engine 5.6; Data Sheet: <http://www.3ds.com/fileadmin/PRODUCTS/SIMULIA/PDF/datasheets/SIMULIA-SEE-56-datasheet.pdf>
- Dassault Systems. (2011, 04 22). *Dassault Systems*. (D. Systems, Ed.) Retrieved 04 22, 2012, from DS Simulia V6R2012x - FACT SHEET: <http://www.3ds.com/fileadmin/PRODUCTS/SIMULIA/PDF/datasheets/SIMULIA-V6R2012x-factsheet.pdf>



- Dassault Systems. (2012, 01). *Dassault Systems*. Retrieved 04 22, 2012, from SIMULIA ExSight Data Sheet:  
<http://www.3ds.com/fileadmin/PRODUCTS/SIMULIA/PDF/datasheets/SIMULIA-ExSight.pdf>
- Dassault Systems. (2012, 10 17). *Dassault Systems Isight Brochure*. Retrieved 10 17, 2013, from Dassault Systems:  
<http://www2.3ds.com/fileadmin/PRODUCTS/SIMULIA/PDF/brochures/simulia-isight-brochure.pdf>
- Davison, R. M. (1998, May). An Action Research Perspective of Group Support Systems: *How to Improve Meetings in Hong Kong*. Hong Kong: City University of Hong Kong.
- Dehning, C., & Wolf, K. (2006). *Why do Multi-Physics Analysis?* NAFEMS Ltd; The International Association for the Engineering Analysis Community.
- Department of Defense, Systems Management College. (2001). *SYSTEMS ENGINEERING FUNDAMENTALS*. Fort Belvoir, Virginia: DEFENSE ACQUISITION UNIVERSITY PRESS.
- Diamond, B. (1993). Extend: A Library-Based, Hierarchical, Multi-Domain Modeling System. *Proceedings of the 25th conference on Winter simulation* (pp. 240-248). ACM.
- Dr. Hartmann, D., & Mahler, M. (2013). Integration of complex 3D-simulations within system simulations using response surface. *NAFEMS CFD in der Systemsimulation Conferenc November 2013*, (pp. 71-79). Bamberg.
- Dr. Ing. Sippel, H., & Dipl. Ing. Niederauer, E. (2010). Was kann Simulation im Produkt-Entwicklungsprozess (PEP) leisten? (What will achieveabel with simulaiton in the product-development-process?). *15.SIMVEC Kongress: Berechnung und Simulation im Fahrzeugbau 2010* (pp. 203-214). Düsseldorf: VDI Verlag GmbH.
- Dr.-Ing. J., F., Dipl.-Ing. S., G., & Prof. Dr.-Ing. H., B. (2010). Ganzheitliche Simulation von mechatronischen Schwingungstilgern für automobile Anwendungen (over all simulation of mechatronic oscilation damper for automotive useage). *NAFEMS Magazin 2/2010*, 16, pp. 22-34.
- Dwarakanath, S., & Wallace, K. M. (22-24 Aug. 1995). Design decision making process: observations from individual and group design experiments. *10th International Conference on Engineering Design (ICED'95)*, 23, pp. 555-560. Heurista.
- Eigner, M., Gilz, T., Hollerith, T., & Zafirov, R. (07. Nov. 2011). Methoden für die modellbasierte interdisziplinäre Produktentwicklung (methods for model-based interdisciplinary product-development). *3. PLM Future Workshop* (p. 26 slides). TU Kaiserslautern: VPE Lehrstuhl für Virtuelle Produktenwicklung.
- Engelson, V. (2000). Tools for Design, Interactive Simulation, and Visualisation of Object-Oriented Models in Scientific Computing. *Linköping Studies in Science and Technology*;

*PhD thesis/Dissertation No. 627.* Linköping university, Sweden: Department of Computer and Information Science.

Fachbach, D. B., & Rosenberger, M. (2010, März). Herausforderungen bei der Implementation von Simulationsdatenmanagement (SDM) in komplexen Entwicklungsumgebungen (Challenge in the implementation of SDM into complex development processes). *NAFEMS MAGAZIN 1/2010(15)*, pp. 26-38.

FAN-LI, Z., LI-PING, C., YI-ZHONG, W., JIAN-WAN, D., JIAN-JUN, Z., & YUN-QING, Z. (2006). MWorks: a Modern IDE for Modeling and Simulation of Multidomain Physical Systems Basen on Modelica. *Modelica 2006, September 4th – 5th* (pp. 725-731). The Modelica Association.

Fernandez-Lopez, M., & Corcho, O. (2004). *Ontological Engineering*. (1st, Ed.) Springer.

Geimer, P. D., Krüger, T., & Linsel, P. (2006). Co-Simulation, gekoppelte Simulation oder Simulatorkopplung? (Co-Simulation, coupled simulation or simulation-coupling?). *O + P Zeitschrift für Fluidtechnik*, 50(11-12), 572-576.

Giese, H., Graf, J., & Witz, G. (1999). *Closing the Gap Between Object-Oriented Modeling of Structure and Behavior*. Münster: Universität Münster.

Giptner, D.-I. (., Moshhammer, D. D., & Panzer, D.-I. D.-I. (2010). Unternehmensübergreifendes multidisziplinäres Simulationsdatenmanagement auf Basis von Open Source Komponenten (company overlapping multi-disciplinary simulation data management founded on Open Source components). *VDI-Bericht Nr. 2107; 15. Kongress SIM VEC Berechnung und Simulation im Fahrzeugbau 2010* (pp. 215-228). Düsseldorf: VDI Verlag GmbH.

Green, G., Kennedy, P., & McGown, A. (2002). Management of multi-method engineering design research: a case study. *Journal of Engineering and Technology Management*, 19(2), 131-140.

Heber, G., & Gray, J. (2005). *Supporting Finite Element Analysis with a Relational Database Backend; Part I: There is Life beyond Files*. Microsoft Research; Advanced Technology Division; Microsoft Corporation. One Microsoft Way; Redmond; WA 98052: April.

Hoog, R. d. (1998). Methodologies for Building Knowledge Based Systems: Achievements and Prospects. In L. J. (ed), *handbook of Expert Systems*. Boca Raton, Florida, USA.

Hui, A., Liping, C., Li, W., & Tifan, X. (2011, 10 18). A Collaborative Platform for Complex Product Development Based on Multi-Domain Unified Modeling and Simulation. *Wuhan University Journal of Natural Sciences*, 16(3), pp. 206-212.

IEEE The Institute of Electrical and Electronics Engineers, I. (1990). IEEE Standard for Developing Software Lifecycle Processes. New York, New York, USA: IEEE The Institute of Electrical and Electronics Engineers, Inc.

- Inc., Siemens Product Lifecycle Management Software. (2001). *Siemens Product Lifecycle Management Software Inc.* Retrieved 04 2012, 18, from [http://www.plm.automation.siemens.com/de\\_de/Images/Siemens-PLM-TEAMCENTER-Integration-For-Doors-fs\\_tcm73-117636.pdf](http://www.plm.automation.siemens.com/de_de/Images/Siemens-PLM-TEAMCENTER-Integration-For-Doors-fs_tcm73-117636.pdf)
- Inc., Siemens Product Lifecycle Management Software. (2009). *NX Nastran User's Guide*. Inc., Siemens Product Lifecycle Management Software.
- Inc., Siemens Product Lifecycle Management Software. (2011). *NX 8 Hilfe NX Nastran User Guide*.
- INCOSE International concile on System Engineering. (2013, 01 02). *A Consensus of the INCOSE Fellows*. Retrieved 01 02, 2013, from <http://www.incose.org/practice/fellowsconsensus.aspx>
- Jenkins, B. (. (2012, 04 01). *Siemens PLM*. Retrieved 04 01, 2012, from Siemens PLM: [http://www.plm.automation.siemens.com/en\\_in/Images/wp\\_best\\_practice\\_series\\_simulation\\_dm\\_060620\\_tcm641-4825.pdf](http://www.plm.automation.siemens.com/en_in/Images/wp_best_practice_series_simulation_dm_060620_tcm641-4825.pdf)
- Katasonov, A., & Jyvässkylä, U. o. (2008, Autumn). *Lecture 6: Requirements Validation and Verification*. Retrieved from [http://www.bilkent.edu.tr/~bakporay/cs\\_413/Lecture6\\_Requirements\\_V&V.pdf](http://www.bilkent.edu.tr/~bakporay/cs_413/Lecture6_Requirements_V&V.pdf)
- Kecskemèthy, A. (31/2007). Mehrkörpersimulation (multi-body-simulation). *ESSENER UNIKATE*.
- Kennedy, P. (2008). *Practical and Scientific Aspects of Injection Moulding Simulation*. Eindhoven: Technische Universiteit Eindhoven.
- Klotzbach, S., Oedekoven, S., & Grassmann, O. (2011). *Optimierung im mechatronischen Entwicklungsprozess (optimization of the mechatronic development process)*. Köln: DMECS Development of Mechatronic Systems.
- Kofman, E., Lapadula, M., & Pagliero, E. (2003). *PowerDEVS: A DEVS-Based Environment for Hybrid System Modeling and Simulation*. Universidad Nacional de Rosario: School of Electronic Engineering, Universidad Nacional de Rosario, Tech. Rep. LSD0306.
- Kondragunta, N. (2010, August 19-20). TEAMCENTER for Simulation Workshop. *Asia Pacific Training*. Japan: Siemens Product Lifecycle Management Software Inc.
- Kossel, R., Claudene, C., & Loeffler, M. (2009, March). Verteilte Systemsimulation mit TISC (distributed system-simulation with TISC). *ASIM-Workshop 2009 Dresden*. Braunschweig: TLK-Thermo GmbH.
- Kothari, C. R. (1990). *Research methodology : methods and Techniques* (2nd ed.). UK: Motilal Books UK.

- Kübler, R. (2000). Modulare Modellierung und Simulation mechatronischer Systeme (modular modeling and simulation of mechatronic systems). *Fortschr. Ber. VDI Reihe 20 Nr. 327*, 148. Düsseldorf, Germany: VDI Verlag GmbH.
- Lennon, T. (November 21, 2007). Model-based design for MECHATRONIC SYSTEMS. *Machine Design*, 42-51.
- Link, K. (2012). The impact of FMI on In-house Tool. *OpenModelica Annual Workshop February 6, 2012*. Linköping University.
- LMS International. (2011). LMS Imagine.Lab SysDM. Leuven, Belgium: LMS International.
- LMS International. (2012, 01 16). *LMS Imagine.Lab SysDM*. Retrieved 01 16, 2013, from <http://www.lmsintl.com/LMS-Imagine-Lab-SysDM>
- LMS INTERNATIONAL. (2012). Math-based plug-and-play software for automotive system design. *Autonomie*. Leuven, Belgium: LMS INTERNATIONAL.
- Mahler, M. (2010). Präsentation at Wieland Werke. *Wieland Werke*. Vöhringen, Germany: not published, only internal use at Siemens SVN.
- Mahler, M. (2012). Framework for System-Simulation based System Engineering Development Methodology. *NAFEMS deutschsprachige Konferenz 2012* (S. 285-288). Bamberg: NAFEMS.
- Mahler, M. (2012). *internal research Paper Daimler interdisciplinary modelbased development process*. Siemens Industry Software GmbH & Co. KG.
- Mahler, M., & Schmidt, R. (2012). Multiphysics Simulation of Moulding Tools for Qualified Designers and Engineers. *NAFEMS European Conference MULTIPHYSICS SIMULATION 16-17 October 2012* (pp. 67-69). Frankfurt Germany: NAFEMS GmbH.
- Mahler, M., Vickers, S., Hasse, A., Traulich, S., Schmerr, A., & Hitzer, H. (2010). Workpackage 4 – Engineering Methodology 2015. *Daimler and Siemens internal Project summary*. Stuttgart, Germany: Siemens Product Lifecycle Management.
- Mammen, D. H.-T. (2012). *Modellierung / Simulation mechatronischer Systeme mit Hilfe der Simulationsumgebung Dymola (modeling / simulation of mechatronic system by using simulation-software Dymola)*. Hella KGaA Hueck & Co. Mannheim: Dassault.
- MathWorks, Inc. (2011). Entwickeln von Modellhierarchien (Development of model-hyrachies). In I. MathWorks, *System- und Algorithmenmodellierung in Simulink*. MathWorks, Inc.
- MathWorks, Inc. (2012, 12 11). datasheet. *Simulink Coder*. MathWorks, Inc.
- McLean, C., Riddick, F., & Lee, Y. T. (2005, January). AN ARCHITECTURE AND INTERFACES FOR DISTRIBUTED MANUFACTURING SIMULATION. *SIMULATION*, 81, 15-32.

- Mellichamp, J. M., & Bengtson, N. M. (1979, May). Simulation environment for designing the dynamic motion behavior of the mechatronic system machine tool. *Interfaces*, 9(3), 87-93.
- MODELISAR consortium. (2010-10-12). *Functional Mock-up Interface for Co-Simulation*. MODELISAR consortium.
- MODELISAR consortium. (January 26, 2010). *Functional Mock-up Interface for Model Exchange MODELISAR (ITEA2 - 07006)*. MODELISAR consortium.
- MODELISAR consortium. (September 30, 2010). *Functional Mock-up Interface for Co-Simulation MODELISAR (ITEA2 - 07006)*. MODELISAR consortium.
- MSC.Software Corporation. (2011). *MSC.Software Homepage*. Retrieved 04 30, 2012, from [http://simcompanion.mscsoftware.com/resources/sites/MSC/content/meta/DOCUMENTATION/9000/DOC9851/~secure/simxpert\\_2011.1\\_doc\\_release.pdf?token=/mq4JRV7j7zhy6wbiDACKCUo32qBGCaFd5VTW1ukLm6SBoAthttsBVrRipLnJoQbwuDsLJQQgV8XX0+KvmgSM9aQt2e8J2jx/Gw/T9W4dhpA3QYd6](http://simcompanion.mscsoftware.com/resources/sites/MSC/content/meta/DOCUMENTATION/9000/DOC9851/~secure/simxpert_2011.1_doc_release.pdf?token=/mq4JRV7j7zhy6wbiDACKCUo32qBGCaFd5VTW1ukLm6SBoAthttsBVrRipLnJoQbwuDsLJQQgV8XX0+KvmgSM9aQt2e8J2jx/Gw/T9W4dhpA3QYd6)
- MSC.Software GmbH. (2012). *MSC.Software Homepage*. Retrieved 04 30, 2012, from <http://www.mscsoftware.com/Germany/Submitted-Content/PDF/SimManager.pdf1.pdf>
- Nowakowski, C., Fehr, J., & Eberhard, P. (2011, 08). Einfluss von Schnittstellenmodellierungen bei der Reduktion elastischer Mehrkörpersysteme (Influence of interface-modelling in the case of reduced elastic multi-body-systems). *at – Automatisierungstechnik*, 512-519.
- Nybacka, M., Törlind, P., Larsson, T., & Johanson, M. (2006, January). DISTRIBUTED REAL-TIME VEHICLE VALIDATION. *2006 ASME International Design Engineering Technical Conferences & Computers and Information In Engineering Conference Philadelphia, Pennsylvania, September 10-13, 2006* (pp. 805-812). Pennsylvania: American Society of Mechanical Engineers.
- Oliver, D. W., Kelliher, T. P., & Keegan, J. G. (1997). *Engineering complex systems with models and objects*. New York: McGraw-Hill, cop.
- OML SysML. (2012, June). *OMG System Modeling Language*. Retrieved 06 01, 2013, from <http://www.sysml.org/docs/specs/OMGSysML-v1.3-12-06-02.pdf>
- Otter, M., & Elmqvist, H. (1995). The DSblock model interface for exchanging model components. *Proc. EUROSIM'95* (pp. 505-510). Vienna: ed. F. Breitenacker and I. Husinsky.
- Poldermann, J. W., & Willems, J. C. (1997). *Introduction to Mathematical Systems Theory: A Behavioral Approach* (1 ed.). Springer.

- Roensch, S. (2005, 12 8). FINITE ELEMENT ANALYSIS: Introduction. *The Finite Element Method: A Four-Article Series, newsletter*. newsletter of the American Society of Mechanical Engineers (ASME).
- Roensch, S. (2005, 12 08). FINITE ELEMENT ANALYSIS: Post-processing. *The Finite Element Method: A Four-Article Series, newsletter*. newsletter of the American Society of Mechanical Engineers (ASME).
- Roensch, S. (2005, 12 08). FINITE ELEMENT ANALYSIS: Pre-processing. *The Finite Element Method: A Four-Article Series, newsletter*. newsletter of the American Society of Mechanical Engineers (ASME).
- Roensch, S. (2005, 12 08). FINITE ELEMENT ANALYSIS: Solution. *The Finite Element Method: A Four-Article Series, newsletter*. newsletter of the American Society of Mechanical Engineers (ASME).
- Ruel, C. (2011). Multi-Physics Simulation Technology in NX. *25th European Workshop on Thermal and ECLS Software*. Noordwijk, The Netherlands: ESTEC.
- Ryan, A. B. (2006, Mar 04). Post-Positivist Approaches to Research. *Researching and Writing your thesis: a guide for postgraduate students*, 12-26. MACE: Maynooth Adult and Community Education.
- Scharff, D., Kaiser, C., Tegethoff, W., & Huhn, M. (2012). Ein einfaches Verfahren zur Bilanzkorrektur in Kosimulationsumgebungen (Easy correction approach for co-simulation-solve-runs). *SIMVEC Berechnung, Simulation und Erprobung im Fahrzeugbau 2012; 16. Kongress* (pp. 581-596). Baden Baden: VDI Verlag GmbH, Düsseldorf.
- Schuh, P. D. (2011, 08 22). *Enzyklopädie der Wirtschaftsinformatik (Enzyklopädie of economical informatic)*. (Oldenbourg Wissenschaftsverlag (Oldenbourg Science Puplicator)) Retrieved August 22, 2011, from <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/informationssysteme/Sektorspezifische-Anwendungssysteme/Produktionsplanungs--und--steuerungssystem/Product-Life-Cycle-Management>
- Sebastien, C., & Ducellier, G. (2006). CAD and FEA integration in a simulation data management environment based on a knowledge based system. *Proceeidngs of TMCE 2006* (pp. 1719-1730). Ljubljana, Slovenia: Orgnizing Committee of TMCE 006.
- Seidel, F., Brunner, D., & Wißuwa, P. D. (Oktober 2009). ZYKLUSZEITEN GESENKT. *Plastverarbeiter*, 10, 99-101.
- Siemens AG. (2011). Drei Fragen an Anton S. Huber (Three questions on Anton S. Huber). *Industry Journal*(1), 29.



- Siemens PLM. (2012, 04 08). *Simulation process management*. Retrieved 04 08, 2012, from Siemens PLM:  
[http://www.plm.automation.siemens.com/en\\_us/Images/10246\\_tcm1023-79849.pdf](http://www.plm.automation.siemens.com/en_us/Images/10246_tcm1023-79849.pdf)
- Siemens PLM Synchronous Technology Part2/2* (2008). [Motion Picture].
- Siemens Product Lifecycle Management Software. (2011). Wightpaper for TEAMCENTER Mechatronics process management. *Mechatronics process management*. Siemens Product Lifecycle Management Software.
- Siemens Product Lifecycle Management Software Inc. (2012). TEAMCENTER 9.1 Simulation Process Management Guide. In *User Guide TEAMCENTER 9.1* (p. 332). Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2008, November). TEAMCENTER overview. Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2011). Advanced Simulation. In S. P. Inc., *NX 8 Hilfe (Help)*. Planto: Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2011). Advanced Simulation User Guide. In S. P. Inc., *NX 8 Help*. Planto: Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2011). Datenkonvertierung. In S. P. Inc., *NX 8 Hilfe*. Planto: Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2011). Motion Simulation. In S. P. Inc., *NX 8 Hilfe*. Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2011). NX 8 Hilfe CAE Advanced Simulation.
- Siemens Product Lifecycle Management Software Inc. (2011). *NX Nastran Installation and Operation Guide*. Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2011). *NX Nastran User's Guide*. Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2012). Getting Started with Mechatronics Process Management. In S. P. Inc., *TEAMCENTER 9.1 User Guide*. Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2012). Getting Started with Product Structure. In S. P. Inc., *TEAMCENTER 9.1*. Siemens Product Lifecycle Management Software Inc.



- Siemens Product Lifecycle Management Software Inc. (2012). System Engineering Guide. In S. P. Inc., *TEAMCENTER 9.1*. Inc., Siemens Product Lifecycle Management Software.
- Siemens Product Lifecycle Management Software Inc. (2012). TEAMCENTER 9.1 My TEAMCENTER Guide. In *TEAMCENTER 9.1 User Guide* (p. 502). Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2012). TEAMCENTER 9.1 Online Help. USA.
- Siemens Product Lifecycle Management Software Inc. (n.d.). NX Nastran Advanced Dynamic Analysis User's Guide. In S. P. Inc., *NX 8 Help*. Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2011). System-driven product development. *White Paper: Managing the development of complex automotive products through a systems-driven process*, 14. USA: Siemens Product Lifecycle Management Software Inc.
- Siemens Product Lifecycle Management Software Inc. (2012). TEAMCENTER 9.1 Behavior Modeling Tool Intergration Guide. In S. P. Inc., *TEAMCENTER 9.1 User Guide* (p. 33). USA: Siemens Product Lifecycle Management Software Inc.
- Stark, R., Beier, G., Wähler, T., & Figge, A. (2010). Cross-Domain Dependency Modelling - How to achieve consistent System Models with Tool Support. *European Systems Engineering Conference (EuSEC)* (p. 14). Stockholm: <http://publica.fraunhofer.de>.
- Sui, J., & Hirshey II, J. (2000). Automobile CVT Shift and Park Mechanism and Powered Lift Gate Simulation Using ADAMS and CATI. *2000 International ADAMS User Conference* (p. 8). Orlando Florida: ADAMS.
- Systems, D. (2012, 04 19). *Dassault Systems*. Retrieved 04 19, 2012, from Multiphysics Digital Lab: <http://www.3ds.com/products/simulia/portfolio/simulia-v6/v6-portfolio/d/realistic-simulation/s/multiphysics-digital-lab-mpdl/p/multiphysics-model-scenario-and-results/?cHash=5eb2a893dc80f2d8177bddb7740a2d97>
- Tian, Y., Yan, Y., Parkin, R., & Jackson, M. (2008). Development of a visualised modeling and simulation environment for multi-domain physical systems. In *Global Design to Gain a Competitive Edge / An Holistic and Collaborative Design Approach based on Computational Tools* (pp. 469-478). Springer-Verlag London.
- Udomphol, T., & Technology, S. U. (2012, 08 05). *Extrusion*. Retrieved from [http://www.sut.ac.th/engineering/metal/pdf/MetForm/04\\_Extrusion.pdf](http://www.sut.ac.th/engineering/metal/pdf/MetForm/04_Extrusion.pdf)
- Ulrich, D. D. (2011). Simulation Data Management - SDM. In *Automotive CAE Companion* (p. CAE Wissen on 19 pages). Alzenau: carsh.training gmbH.

- van Beek, D. A., Rooda, J. E., Engell, S., & Zaytoon, J. (2000). Multi-domain modeling, simulation, and control. *4th International conference on automation of mixed processes: hybrid dynamic systems*, (pp. 139-146). Dortmund; Germany.
- Vasaiely, P. (2009, 08 24). Interactive Simulation of SysML Models using Mouldelica. *Bachelor Thesis*. Hamburg: Hochschule für Angewandte Wissenschaften Hamburg.
- VDA. (2008). *Simulation Data Management*. Verband der Automobilindustrie, joint VDA and ProSTEP iViP Association Project Group SimPDM. Westendstr. 61; 60079 Frankfurt: Verband der Automobilindustrie.
- Vrinat, M. (2009). *Simulation Data & Process Management - Vendor Scorecard*. 44 Summit Road, Suite 101, Riverside, CT 06878: Collaborative Product Development Associates, LLC.
- Vyatkin, V., Hanisch, H.-M., Cheng, P., & Chia-Han, Y. (2009). Closed-Loop Modeling in Future Automation System Engineering and Validation. *IEEE transactions on systems, man and cybernetics. Part C, Applications and reviews.*, 39, 17-28.
- Vyatkin, V., Hanisch, H.-M., Cheng, P., & Chia-Han, Y. (2009, Feb). Closed-Loop Modeling in Future Automtion System Engineering and Validation. *Man & Cybernetics: Part C - Applicatoiins & Reviews*, 39(1), 1-12.
- Wenzel, H., Gondhalekar, A., Balachandra, L., Guenov, M., & Nunez, M. (2010). Automated generation of Isight-Models through a neutral workflow description. *SIMULIA Customer Conference*. Paris: Dassault Systems SIMULIA and Cranfield University.
- Wikipedia. (2011, Juni 01). *Wikipedia Product Lifecycle Management*. Retrieved Juni 01, 2011, from [http://de.wikipedia.org/wiki/Product\\_Lifecycle\\_Management](http://de.wikipedia.org/wiki/Product_Lifecycle_Management)
- Wikipedia. (2011, May 21). *Wikipedia Produktdatenmanagement*. Retrieved Juni 01, 2011, from <http://de.wikipedia.org/wiki/Produktdatenmanagement>
- WIKIPEDIA. (2012, 08 05). *Extrusion*. Retrieved from <http://en.wikipedia.org/wiki/Extrusion>
- WÜRTELE, D.-I. M., LANGE, D.-I. C., & HUNGERKAMP, D.-I. T. (2004). Plastifizieren in kürzester Zeit (plasticfication in shortest time). 06.
- Wuttke, F., Bohn, M., & Suyam-Welakwe, N.-A. (2011). Early Robust Design Approach for Accelerated Automotive Innovation Processes. In Cavallucci Denis, R. Guio, & G. Cascini (Ed.), *Building Innovation Pipelines through Computer-Aided Innovation. 4th IFIP WG5.4 Working Conference, CAI 2011 Strasbourg, France, June/July 2011*, p. 24. Strasbourg, France: Springer.
- Yin, R. K. (1984). *Case Study Research: design and methods*. Beverly Hills: Sage Publication.

- Zaeh, M. F., & Baudisch, T. (2003). *Simulation environment for designing the dynamic motion behaviour of the mechatronic system machine tool*. Institute for Machine Tools and Industrial Management, Technische Universität München. Munic: Proceedings of the Institute for Machine Tools and Industrial Management, Technische Universität München.
- Zehetner, D., Wenpu Lu, Watzenig, D., & Bernasch, D. (2012). Co-Simulation und Modellbibliothek - Eckpfeiler für einen modernen Systementwicklungsprozess (Co-Simulation and model-library - Foundings for moderen system-development-processes). *SIMVEC Berechnung, Simulation und Erprobung im Fahrzeugbau, 16. Kongress in Baden-Baden* (pp. 597-607). Düsseldorf: VDI-Verlag GmbH.
- Zimmer, D. (2010). Equation-Based Modeling of Variable-Structure Systems; PhD thesis. *A dissertation submitted to the Swiss Federal . Schweiz*, Institute of Technology, ETH Zürich: ETH Zürich.

## 9 APPENDIX

### 9.1 APPENDIX A SYSTEMATIC LITERATURE REVIEW

The literature research will be based on searching key words. In order to get a speedy and efficient search, the literature search should be organised in more effective way. Therefore, the search words will be organised into a search algorithm. The search algorithm will put the search words in a logical dependency. This logical dependency achieves a faster search result. The different databases use different logical functions. So, the search algorithm has to be optimised for each literature database. In Table 9-1 Search Functions, the search words, the logical search function and the two different search algorithms, optimised for the literature search databases, are listed:

**Table 9-1 Search Functions**

<b>Search Words</b>	<b>Logical Search Function</b>	<b>EBSCO database Search Function</b>	<b>Sage Journal Online database Search Function</b>
<ul style="list-style-type: none"> <li>- system-simulation</li> <li>- system-Simulation</li> <li>- mechatronic-system-simulation</li> <li>- mechatronic-simulation</li> <li>- co-simulation</li> <li>- multi-disciplinary-simulation</li> <li>- multi-domain-simulation</li> <li>- parallel-simulation</li> <li>- system-engineering simulation</li> </ul>	system or mechatronic or co or parallel or (multi and domain or discipline) and simulation	TI simulation and (TI system or TI mechatronic or TI co or TI parallel or TI multi domain or TI multi discipline)	(simulation and system) or (simulation and mechatronic) or (simulation and co) or (simulation and parallel) or (simulation and multi-domain) or (simulation and multi-discipline)
<ul style="list-style-type: none"> <li>- system-simulation-data-management</li> <li>- co-simulation-data-management</li> </ul>	(system and simulation) or (co and simulation) and (data and management)	TI data management and (TI system-simulation or TI co simulation)	(simulation and system) or (simulation and mechatronic) or (simulation and co) or (simulation and parallel) or (simulation and multi domain) or (simulation and multi discipline)

<ul style="list-style-type: none"> <li>- co-simulation-interface</li> <li>- simulation-interface</li> <li>- multi-disciplinary-simulation-interface</li> </ul>	(co and simulation) or (multi and disciplinary and simulation) or simulation and interface	TI interface and (TI co simulation or TI multi disciplinary simulation or TI simulation)	(interface and co simulation) or (interface and multi disciplinary simulation) or (interface and simulation)
<ul style="list-style-type: none"> <li>- multi-domain-modelling</li> <li>- multi-domain-pre-processing</li> </ul>	(multi and domain) and (modelling or pre-processing)	TI multi domain and (TI modelling or TI pre-processing)	(multi-domain and modelling) or (multi-domain and pre-processing)
<ul style="list-style-type: none"> <li>- simulation-processes</li> <li>- simulation middleware</li> <li>- simulation-automation</li> </ul>	simulation and (processes or middleware or automation)	TI simulation and (TI processes or TI middleware or TI automation)	(simulation and processes) or (simulation and middleware) or (simulation and automation)

Nevertheless, the literature search results in a protracted list of literature. This quantity of literature has to be filtered, as discussed in Section 9.1.1.

### 9.1.1 SELECTION OF LITERATURE

The literature research identified 39,060 documents. This is a huge number of documents which cannot be checked manually. Therefore, a filter is required.

A filter should be the publication date of the documents. Computer-aided engineering is a relatively new discipline. The research project will work on multi-disciplinary simulations which are newer than computer-aided engineering. Documents older than 20 years would not include new information. So, a filter could be set with a maximum document age of 20 years. Therefore, the earliest year of population should be 1992. (The main literature search was carried out in 2012).

A second filter could be the scholarly or academic filter. The documents should be focused and based on academic work.

The filters were combined with the search algorithms. In some cases, the date or scholarly filter significantly reduced the number of documents. In such cases, the filters were not used. In summary, the filters could achieve effective reductions of the identified amount of literature.

### 9.1.2 QUALITY AND RELEVANCE OF THE LITERATURE

814 literature documents were identified by the search algorithms and filters. Nevertheless, this number was still too high and had to be reduced. A fast review of the research results showed that literature was included where simulations were used to clarify technical operations, or where research projects were done with other objectives in mind. This kind of data was not relevant for this research. A filtering of the research result, based on its relevance to the research topic and aims, would be helpful. Therefore, the identified literature was evaluated. This kind of filtering was focused on relevance and brought the literature in line with the research.

The evaluation of the literature had to be completed manually. This kind of evaluation was not possible in one step. It would have taken a long time to evaluate this amount of literature. So, a two-step filtering was required:

1. The first step was to evaluate the literature titles. Literature which did not fit the criteria, discussed below, would be filtered. If the titles did not give enough information, they would be set aside for the next step.
2. The second step was to evaluate the literature abstract. The criteria for the evaluation are discussed in the following. An evaluation and rating of the relevance for the research project will also be produced.

The filtering criteria are used to check for a high relevance quotient of the literature for the research project. These criteria can be used for the filtering process of the first step:

1. Is the main theme of the content relevant for simulation themes?
2. Is the content relevant for the connection of different simulation tools?
3. Is the content relevant for the management of simulation data?
4. Is the content relevant for simulation processes?

The filtered literature includes literature with differing relevance to the research project. This literature has to be evaluated and rated according to its relevance. This rating can be seen as a quantitative rating of the literature in response to the research project. For a quantification and rating, a rating level of the relevance will be necessary. Quantification-criteria are necessary. The criteria will represent relevant themes that are appropriate for the research. The criteria are as follows:

1. multi – disciplinary – simulation – data – management
2. multi – disciplinary – simulation – process – management
3. multi – disciplinary – simulation – interfaces - management
4. multi – disciplinary - post-process - management
5. multi – disciplinary - pre-process – management
6. multi - disciplinary – simulation - time – management
7. multi – disciplinary – middleware – concept based on PLM

Each document has to be rated according to these criteria. The criteria were measured in three levels: high (1) – low (0) – nothing (-1). ‘High’ means the literature addresses the theme and ‘nothing’ means the theme is not addressed. The rating of the criteria can be compiled as a sum. The sum represents the quantification of relevance of the literature to the research. The lowest possible level was -7 and the highest level was 7. In Figure 9-1, the distribution of the quantification to the amount of literature is presented. In this figure, the blue colour represents the number of documents for each rating and the red for the number of documents, starting with the lowest ranking.

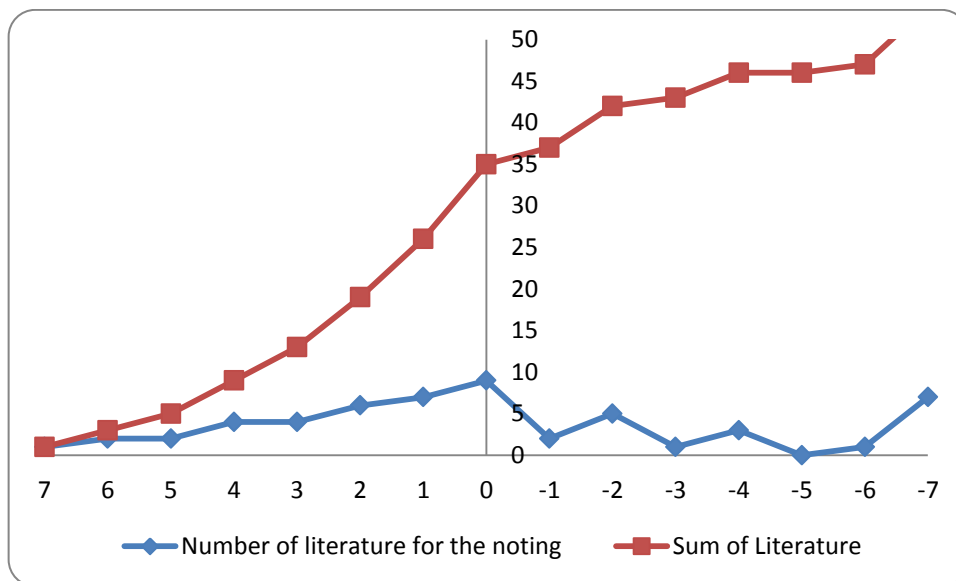


Figure 9-1 Relevance of the Literature

Figure 9-1 Relevance of the Literature should help to identify the most relevant literature as well the ideal amount of literature. Based on this kind of key word search and filtering, the ranking of the literature should relate to the research topic. For an initial overview of the research topic, the literature with rating levels 7,6,5,4 and 3 will be reviewed. The number of the documents with those rating levels is 13. These documents will be listed and synthesised as follows:

- A Collaborative Platform for Complex Product Development based on Multi-Domain Unified Modeling and Simulation by Ai, Hui; Chen, Liping; Wan, Li; Xiong, Tifan. Wuhan University Journal of Natural Sciences, June 2011, Vol. 16 Issue: Number 3 p. 206-212; Database: EJournal.

Ai, Chen, Wan & Xiong (2011) include an interesting discussion about a system-architecture to support multi-domain modelling and simulation. This system-architecture summarises CAD, CAE (computational aided engineering), CAPP (computer aided process planning), CAM (computational aided manufacturing), SCM (supply chain management), ERP (enterprise resource planning) etc. Unfortunately, the paper does not provide a discussion at a level to shape a functioning framework



that supports and manages multi-disciplinary simulation data. The paper focuses on a discussion of a unified management of product knowledge, a comprehensive software tool and system architecture. However, the paper could help to discuss a theory of multi-disciplinary simulation data management.

- Integrated Development of a Visualised Modeling and Simulation platform for Multi-domain Systems by Tian, Yongli and Yan, Yunhui; Journal of Computer Aided Design and Computer Graphics; 2008, 20(4):526-531; Database; British Library Document Supply Centre Inside Serials & Conference Proceedings.

This paper was not written in English or German. However, the paper was also published in English (Tian, Yan, Parkin & Jackson , 2008).

Tian, Yan, Parkin & Jackson (2008) introduce “the development of a software environment for both visualised modeling and simulation of mechatronic multi-domain systems.” This new environment is called Vimola. It uses the Modelica language base. The Modelica language provides the possibility to generate a hybrid modelling platform for multi-domain physical system. Modelica is an open EBS software base and Vimola is an integrated development environment for modelling and simulation of multi domain physical systems. In the research project, EBS is considered a partial solution of a multi-disciplinary simulation. This kind of simulation discipline will be used in the case examples.

- Development of Hybrid Modeling Platform for Multi-Domain Physical System by Yizhong, W. Journal of Computer Aided Design and Computer Graphics; 2009; 18(1): 120-124; Database: British Library Document Supply Centre Inside Serials & Conference Proceedings.

This paper was not written in English or German. Only the title and abstract is available in English. In order to understand this document, an investment in translation had to be done. The abstract of the document gave information about its content. The document content is the MWoks-tool implementation to the Modelica-based hybrid modelling platform for a multi-domain physical system. Modelica is an open EBS software base and MWoks is an integrated development environment for modelling and simulation of multi domain physical systems based on Modelica (FAN-LI, LI-PING, YI-ZHONG, JIAN-WAN, JIAN-JUN, & YUN-QING, 2006). This helps to understand EBS in relation to system-simulation. However, this document is not important for the research project because this content will be covered in (Tian, Yan, Parkin & Jackson , 2008).

- Multi-Domain Modeling, Simulation and Control by van Beek, D. A.; Rooda, J. E.; International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems, (ed. Engell, S.; Kowaleswski, S.; Zaytoon, J.) 139-146 Conf: Conference; 4<sup>th</sup>; Dortmund, Germany Sep 2000  
Database: British Library Document Supply Centre Inside Serials & Conference Proceedings

Van Beek, Rooda, Engell & Zaytoon (2000) discuss the use of EBS tools and the “interaction between simulation models written in different languages” as a suite for “modeling, simulation and control of systems from different application domains” The Chi project, which works on a multi-domain modelling and simulation language, are within the focus of this paper. The Chi language is evaluated, based on the Modelica language. Both languages are EBS languages and provide possibilities to build models with different physical domains such as chemical or mechanical. Such EBS languages and tools provide the possibility to simulate multi-domains. However, in the research project EBS is seen as a partial solution of multi-disciplinary simulation. This kind of simulation discipline is used in the case examples.

- Multi-Domain and Simulation of Mechatronic Systems by Bharadwaj, A. S.; Actuators Automotive Electronics Series PT-74 (ed. Jurgen, R. K.) Progress in Technology; 1998, 74:369-374 Database British Library Document Supply Centre Inside Serials & Conference Proceedings

Bharadwaj (1998) focuses on the presentation of a multi-domain simulation solution for a pressure control solenoid system. This can be considered as a case example for managing the data of multi-disciplinary simulation examples. Unfortunately, the paper includes not detailed information about the managed data. Nevertheless, if required, the document will be used in discussions of the case examples.

- Extend: A Library-Based, Hierarchical, Multi-Domain Modeling System by Bob Diamond. Winter Simulation Conference; 1993:40 Database British Library Document Supply Centre Inside Serials & Conference Proceedings

Diamond (1993) “presents an overview of Extend and discusses its use for both continuous and discrete event modelling”. The use of EBS languages is in the focus of this paper. EBS languages provide the possibility to build models with different physical domains such as chemical or mechanical. However, in the research project EBS is seen as a partial solution of a multi-disciplinary simulation. Additionally, the paper is not up-to-date because the publication date was 1993 and the content is more relevant for EBS languages and tools than for the support of multi-disciplinary simulation data.

- Integration of CAD and FEA (Finite Element Analyses) Data in a PDM System: Definition of a Step Simulation Data Management Schema; Charles, S.; Eynard, B.; in IMACS world congress (ed. Borne, P.) 2005: Paper T6-I81-0302 Conf: 17<sup>th</sup>; Paris; Jul 2005 Database British Library Document Supply Centre Inside Serials & Conference Proceedings.

Sebastien & Ducellier (2006) “presents the development of a Simulation Data Management schema linked with a KBE system. The objective of the research work is to define a full environment aiming to enhance collaboration between design and analysis activities in a PDM approach.” The paper is not focused on multi-disciplinary simulation but on the data management of simulation data in the context of product data. SDM (Simulation Data Management) and PDM interaction is a young discipline. The new framework will improve SDM and PDM interaction. So, the paper could help to discuss theory of multi-disciplinary simulation data management.

- Designing Spacelab's Data Management System with Simulation by Mellichamp, Joseph M.; Bengtson, Neal M.; Interfaces, May 1979, Vol. 9 Issue: Number 3 p87-93, 7p; Database: E-Journals

Mellichamp & Bengtson (1979) is not up-to-date. The paper describes a multi-disciplinary simulation use case. This use case is one of the earliest use cases for multi-disciplinary or multi-domain simulations. Due to the age and history of multi-disciplinary and multi-domain simulations, this paper does not represent the present day and is not relevant for the research project. Additionally, the paper specialises in space shuttles.

- Simulation Environment for Designing the Dynamic Motion Behaviour of the Mechatronic System Machine Tool by Zaeh, M F; Baudisch, T. Proceedings of the Institution of Mechanical Engineers – Part B – Engineering Manufacture (Professional Engineering Publishing); Jul 2003; Vol. 217 Issue 7; p1031-1035, 5p; Database: Business Source Complete

Zaeh & Baudisch (2003) describe “a new method for setting up a model for the mechatronic system machine tool, with special emphasis on the design of its motion dynamics.” The paper is focused on the feasibility and information flow of models taking place in a virtual prototype but doesn't include the management of used data. However, the paper discusses some requirements of companies to generate virtual prototypes. So, the paper could help to discuss a theory of multi-disciplinary simulation data management.

- Closed-Loop Modeling in Future Automation System Engineering and Validation by Vyathkin, Valeriy; Hanisch, Hans-Michael; Cheng Pang; Chia-Han Yang; IEEE Transactions

on Systems; Man & Cybernetics: Part C – Applications & Reviews, Feb2009, Vol. 39 Issue 1, p17-28, 12p, Database: Business Source Complete.

The focus of the paper is a discussion about system engineering and validation as well as verification in the context of system engineering and simulation. It explains that the “proposed ideas can be used to extend the existing engineering support software tools, or as a foundation of new toolsets, by combining system design with the closed loop validation and verification capability.” The paper is not focused on multi-disciplinary simulations. Still, the paper could help to discuss a theory of multi-disciplinary simulation in the context of system engineering.

- An Architecture and Interfaces for Distributed Manufacturing Simulation by Charles Mc Lean; Frank Riddick; Y. Tina Lee; Simulation, January 2005, vol. 81,1 pp. 15-32.

McLean, Riddick & Lee (2005) is focused on “an overview of a neutral reference architecture and data model for integrating distributed manufacturing simulation systems with each other [...]”. The reference architecture is an architectural concept of a simulation middleware. This high level simulation middleware architecture includes the interfacing of the co-simulation runtime infrastructure and the legacy of simulation execution engines. This paper could help to discuss an architectural concept of the new framework.

Also, the need for a data management system is discussed in this paper. This data model should offer the possibility of storing and managing the required data. However, there will not be an in-depth discussion on data management.

- PowerDEVS: a Tool for Hybrid System Modeling and Real-time Simulation by Federico Bergero; Ernesto Kofman; Simulation, January 2011; vol. 87; 1-2; pp113-132; first published on April 28, 2010

Kofman, Lapadula & Pagliero (2003) discuss “the main features of the PowerDEVS software.” The article also illustrates its use with some examples showing its simplicity and efficiency. PowerDEVS uses an EBS language as the simulation base. However, in the research project EBS is only seen as a partial solution of a multi-disciplinary simulation.

The systematic literature research could not identify any document that takes the same viewpoint as the research project. However, a lot of literature is available that focuses on subcategories of the research project. These kinds of documents could be used, for example, in scene setting.

So far, a systematic literature review does not provide an effective base for the literature review. Instead it lends itself to setting the scene of the research project. In summary, the following points have to be reviewed:

- Multi-disciplinary simulation
- Simulation data management
- Data management of multi-disciplinary simulations

## 9.2 APPENDIX B WORK PLAN

Project	process	activities	tasks
	What concept could multi-disciplinary-simulation support in a general way?	<ul style="list-style-type: none"> <li>1.1: Sample multi-disciplinary simulation case examples</li> <li>1.2: Data acquiesce of the simulation data management possibilities of the PLM system                             <ul style="list-style-type: none"> <li>1.2.1: Data sampling of existing data management systems</li> <li>1.2.2: Evaluation of the data management systems</li> <li>1.2.3: Data sampling of simulation data management of a selected data management system</li> <li>1.2.4: Data acquiesce of the simulation data management technology</li> </ul> </li> <li>1.3: Analyze the case examples simulation authoring tools interaction.                             <ul style="list-style-type: none"> <li>1.4: Conceptualization of the new framework.                                     <ul style="list-style-type: none"> <li>1.4.1: architectural new framework approach</li> <li>1.4.2: detailed new framework approach</li> <li>1.4.3: impact analyses of the case example data management with the new framework concept</li> </ul> </li> </ul> </li> </ul>	
		<ul style="list-style-type: none"> <li>2: Conceptualization of a useful supporting framework for the cooperation of simulation models generated by multiple simulation authoring tools.                             <ul style="list-style-type: none"> <li>2.1: Analyze the case examples multi-disciplinary simulation process.</li> <li>2.2: Analyze the case examples ex- and import during the multi-disciplinary simulation process.</li> </ul> </li> <li>3: Conceptualization of a useful supporting framework for the interaction of multiple system and/or product describing structures.                             <ul style="list-style-type: none"> <li>3.1: Analyze the case examples system and product descriptions.</li> </ul> </li> <li>4: Conceptualization of a useful supporting framework to relate data and information of different development-disciplines taking part in a product development process.                             <ul style="list-style-type: none"> <li>4.1: Analyze the case examples data and information relations.</li> </ul> </li> </ul>	
	systematic literatur research methods and methodology		
	RD1 - Formular		
	Writing theoretical base		
	Writing case examples		
	Writing conceptualization		
	Writing methods & methodology, systematic literatur research		
	Writing Paper		
	Working on employee invention		
	Overworking the writing		
	Finalization of the PHD Thesis		
	Hand in thesis and Viva Voce		
	Q2/2011 Q3/2011 Q4/2011	Q1/2012 Q2/2012 Q3/2012	Q1/2013 Q2/2013 Q3/2013 Q4/2013 Q1/2014 Q2/2014



### 9.3 APPENDIX C ENHANCED GRAPHICS OF THE PHD THESIS



Figure 4 6 Standard Simulation Data Model

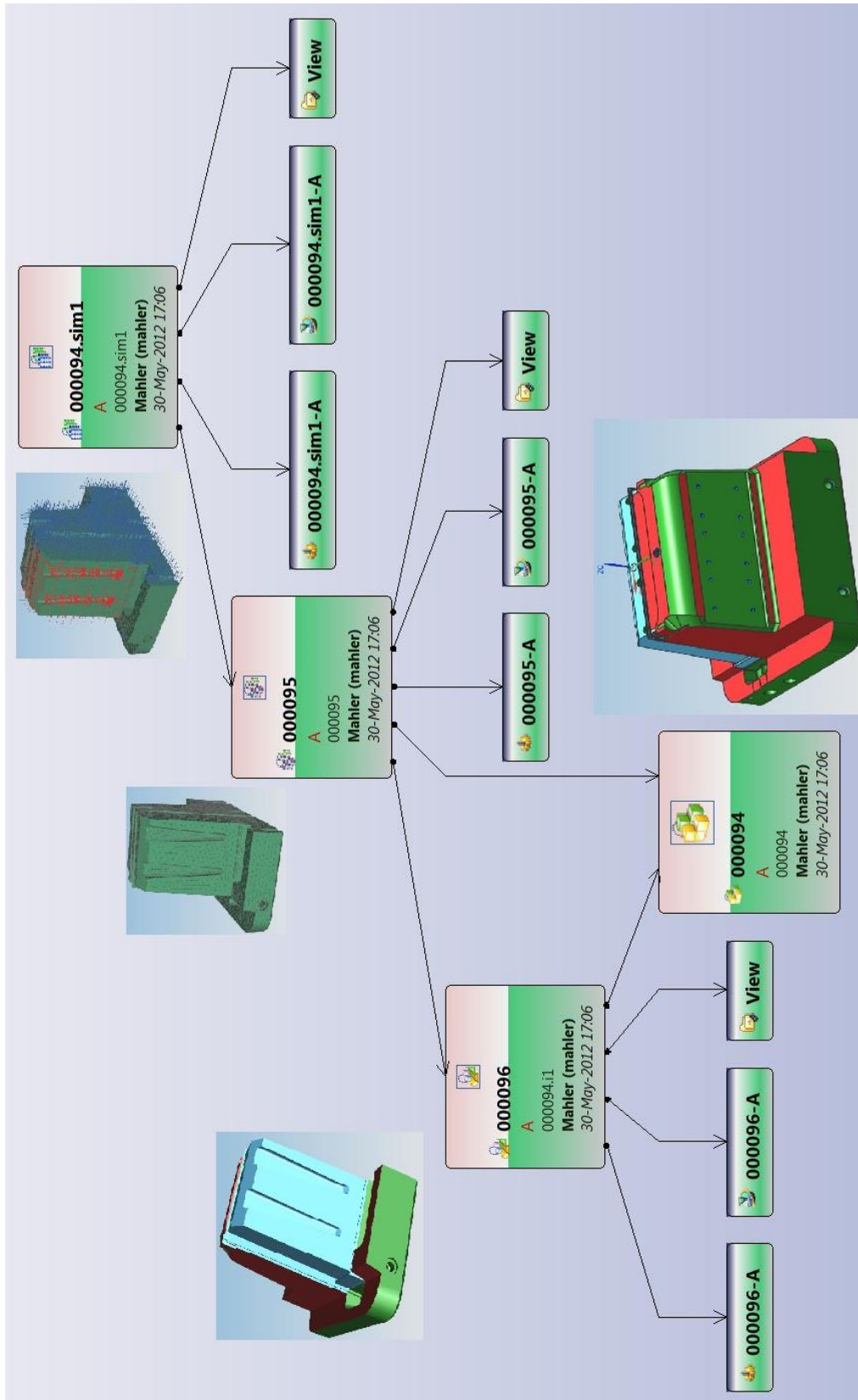


Figure 4 7 Simulation Data Structure Example

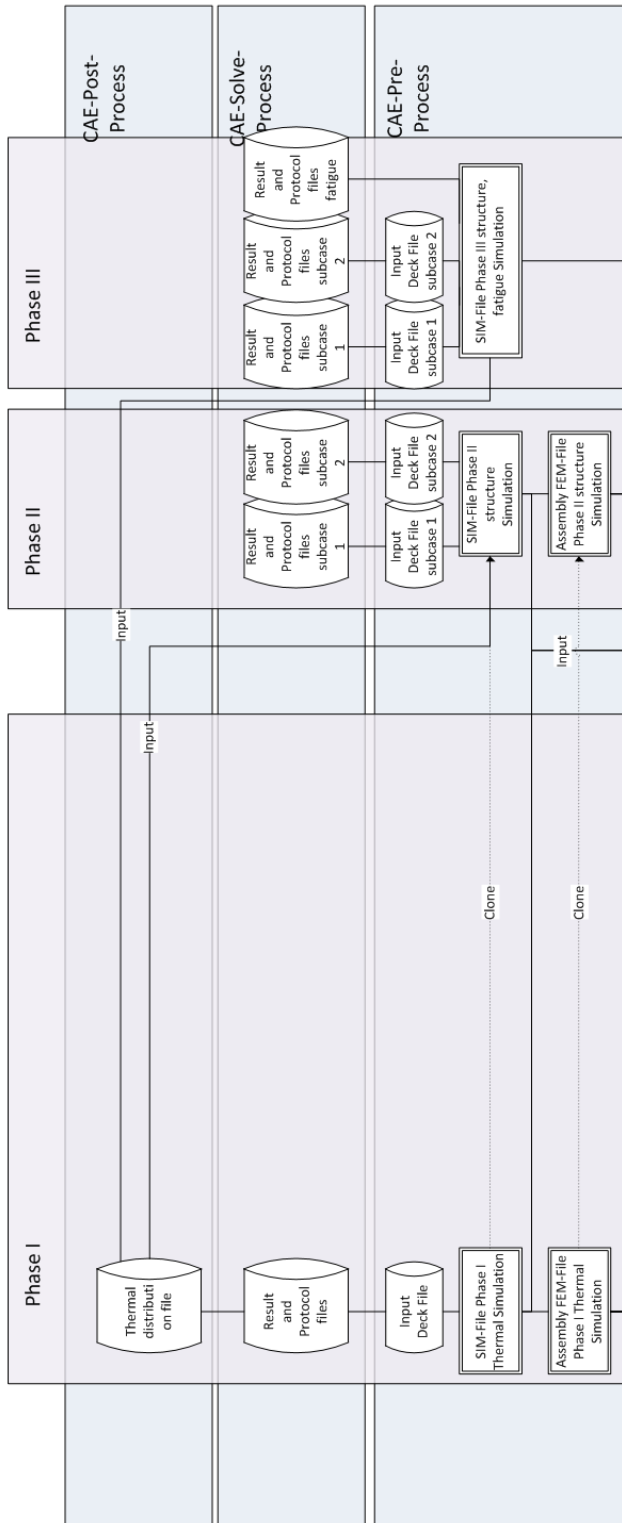


Figure 5 12 Simulation Structure of Case 1

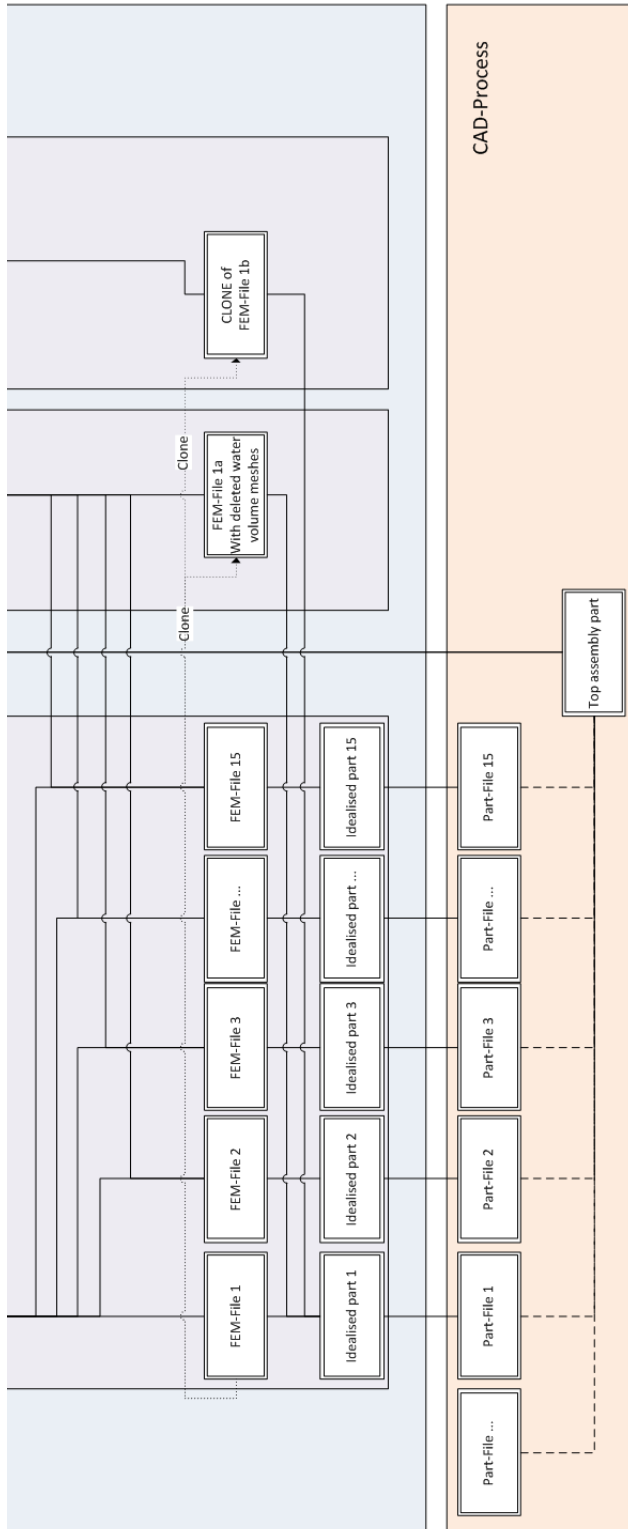


Figure 5 12 Simulation Structure of Case 1

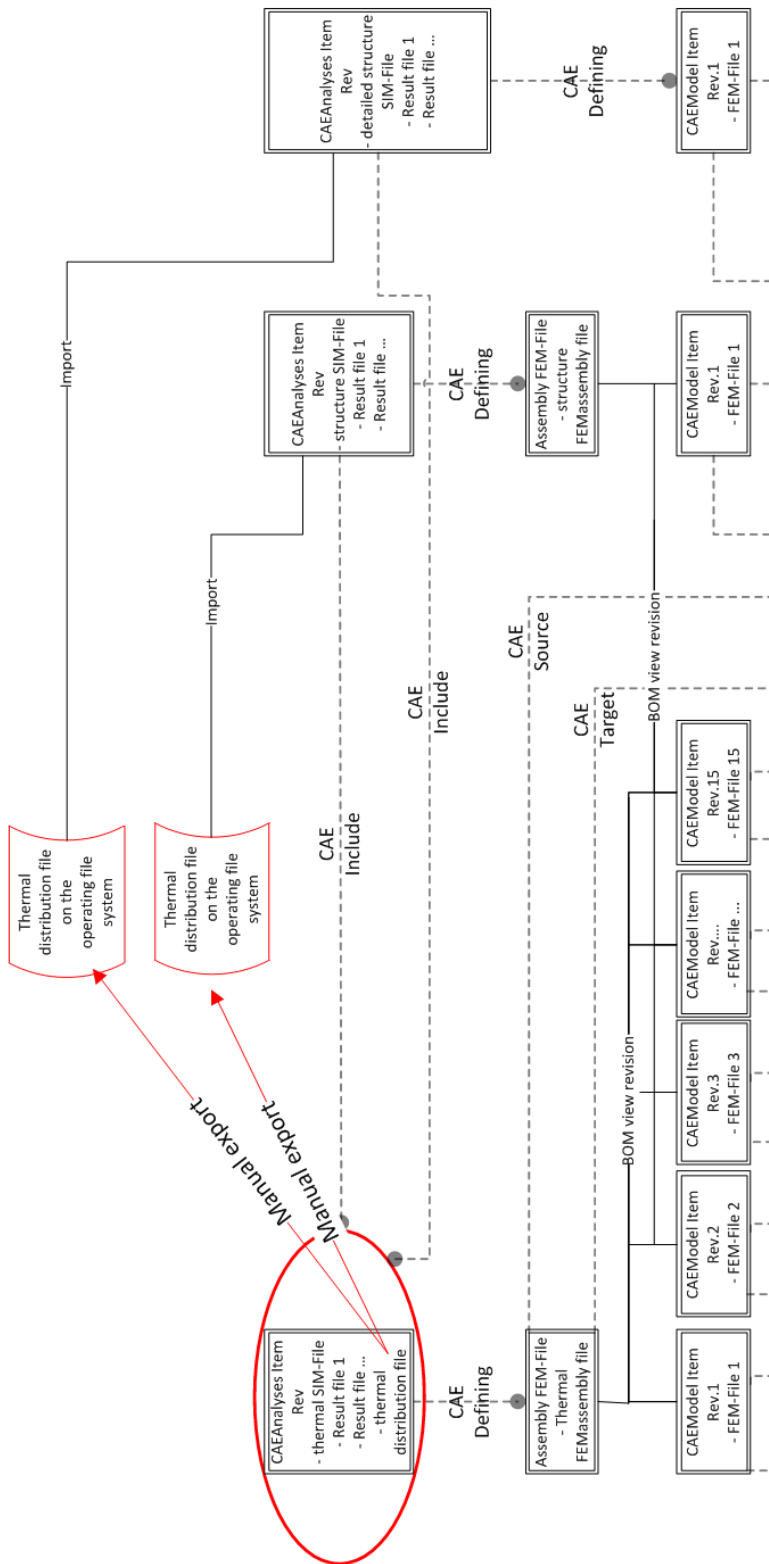


Figure 5 15 Metadata Structure of Case 1 Phase 1, 2 and 3

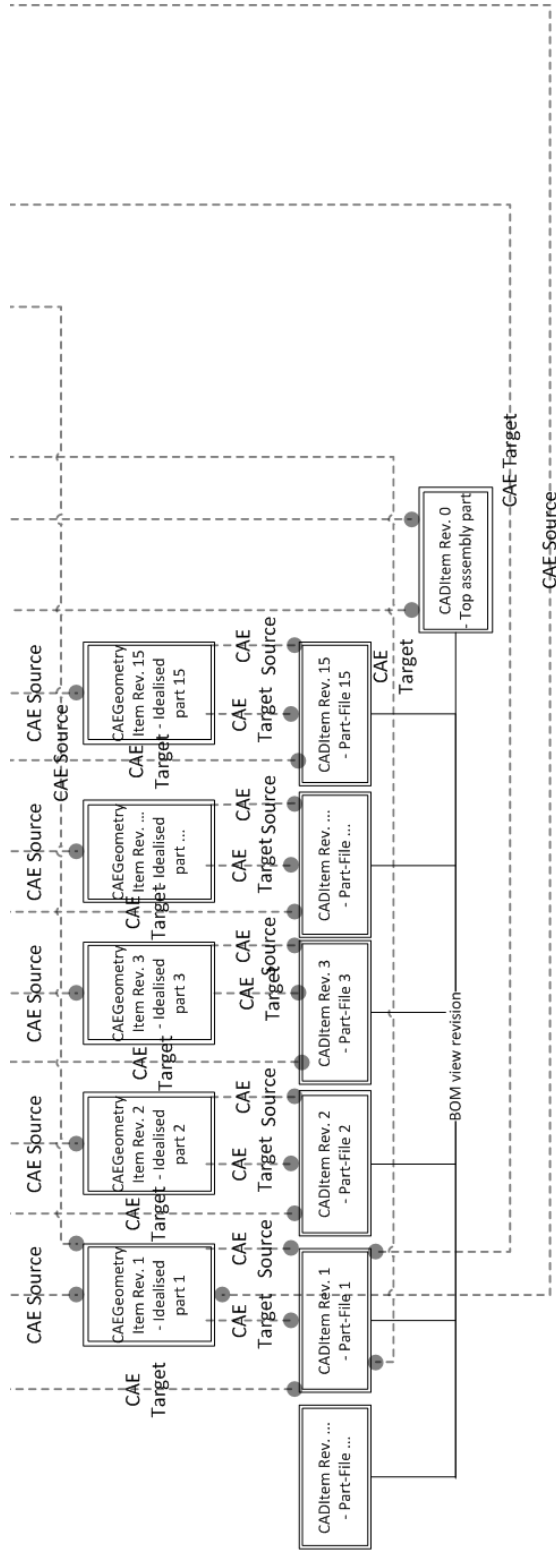


Figure 5 15 Metadata Structure of Case 1 Phase 1, 2 and 3



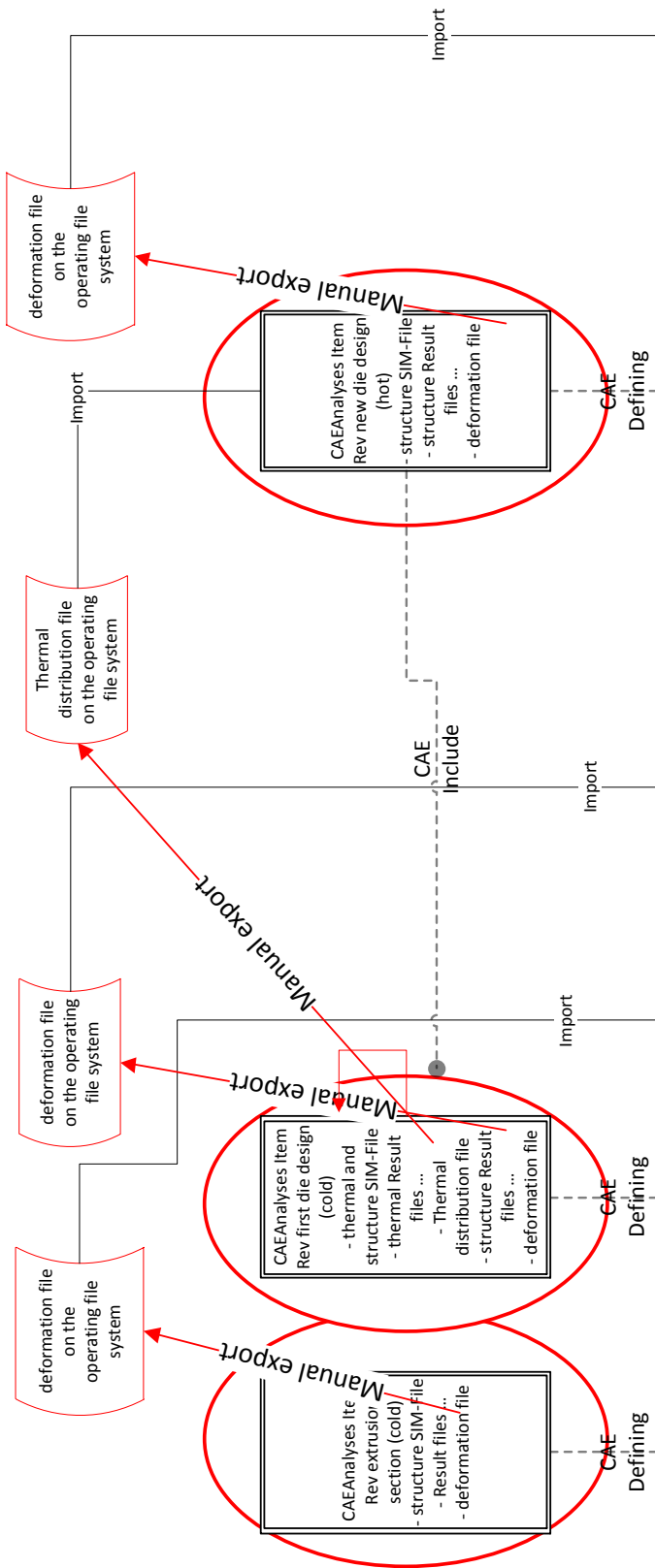


Figure 5 21 Metadata Structure of Case 2

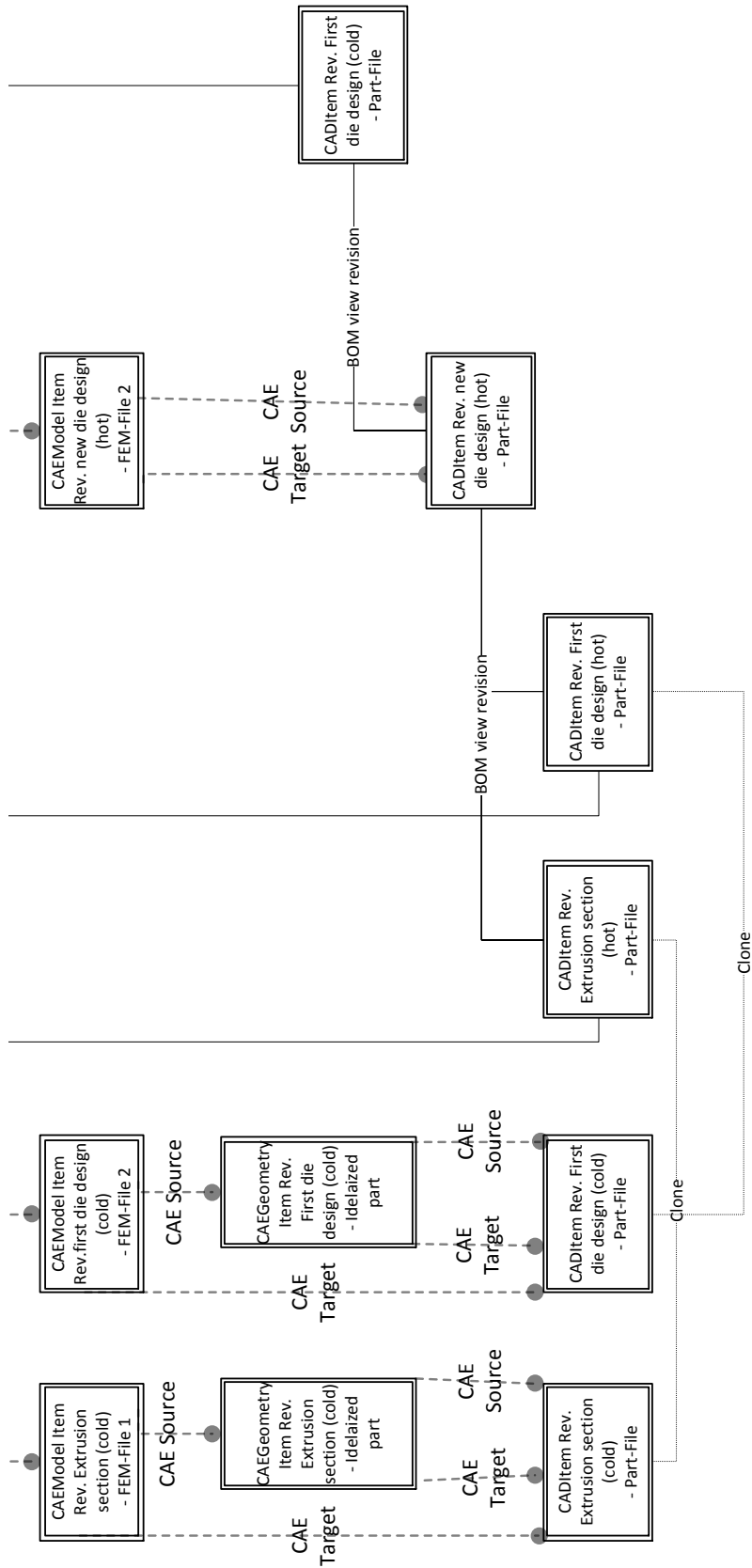


Figure 5 21 Metadata Structure of Case 2

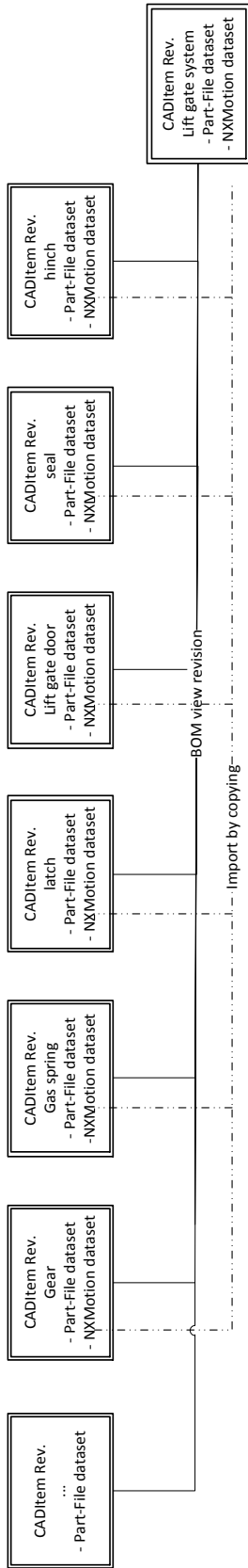


Figure 5 28 Rigid Body Architecture of Case 3

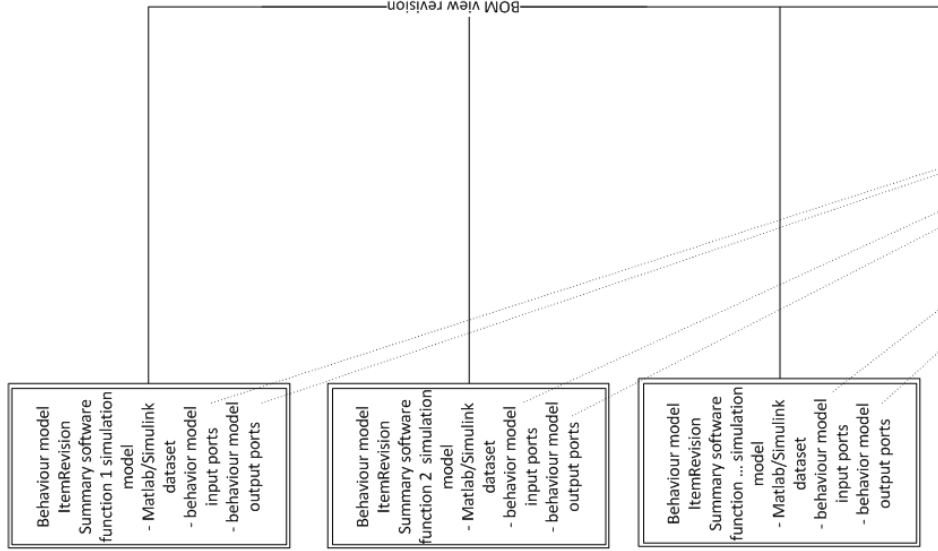


Figure 5 31 Behaviour Model Structure of Case 3

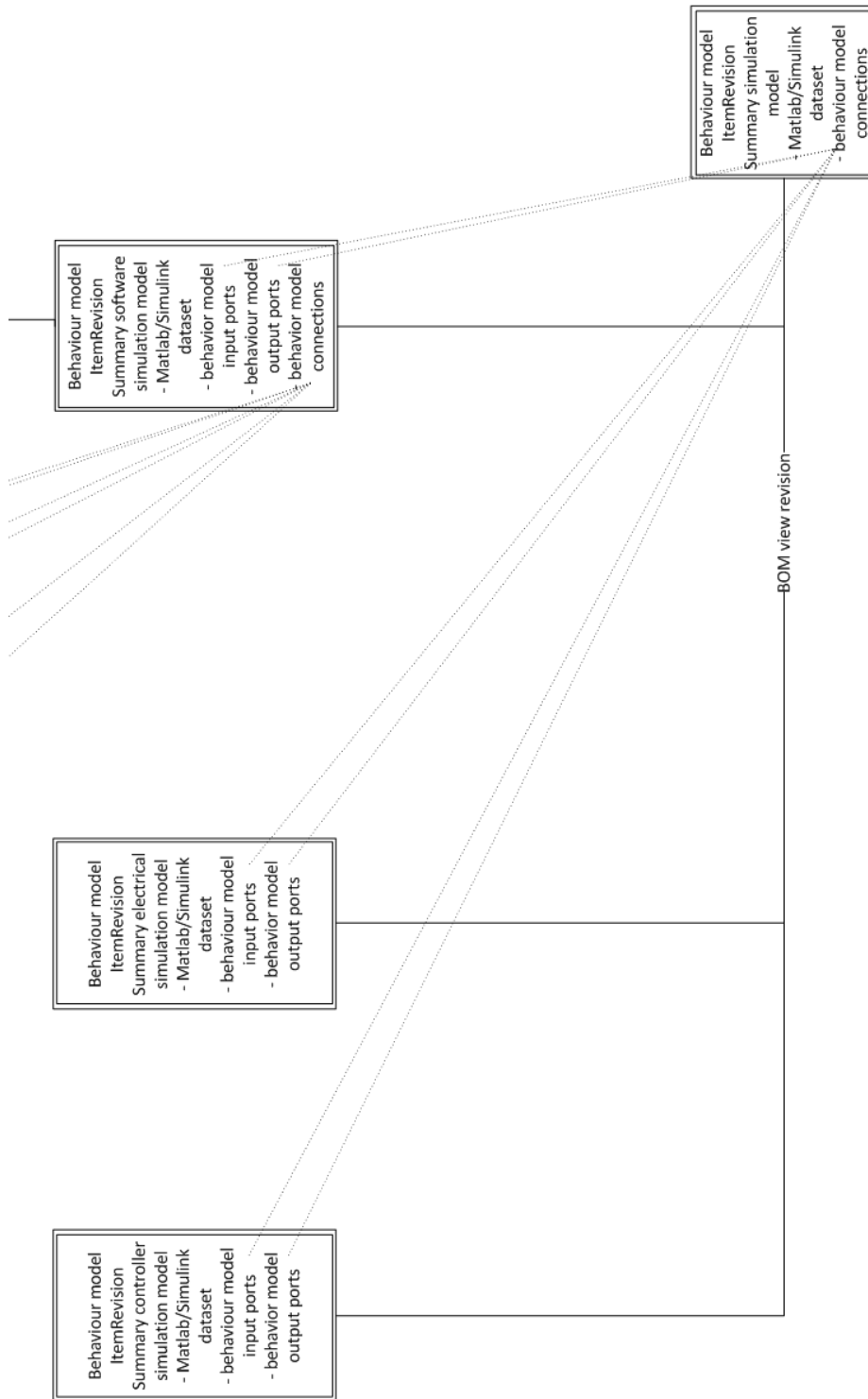


Figure 5 31 Behaviour Model Structure of Case 3

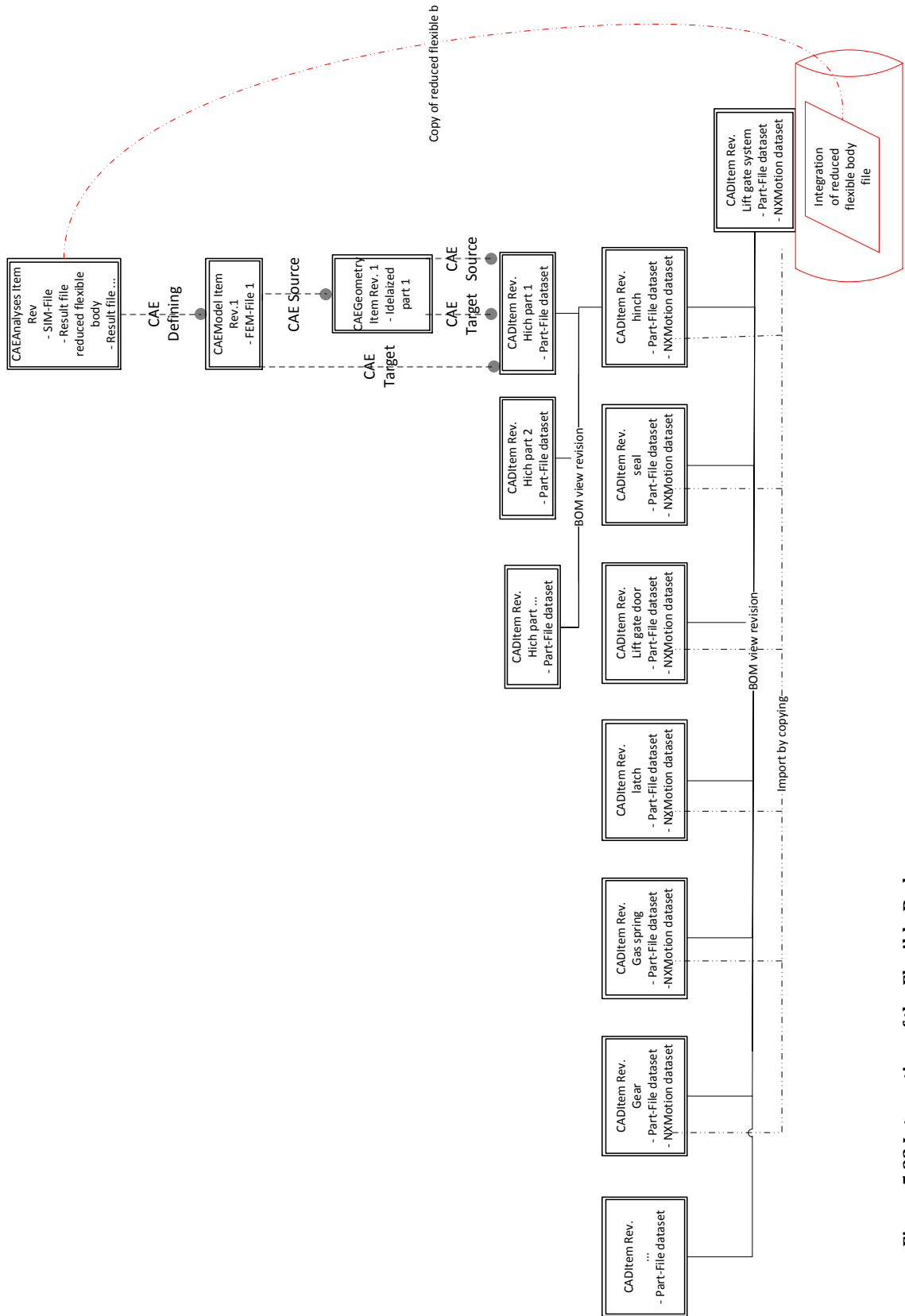


Figure 5 32 Integration of the Flexible Body

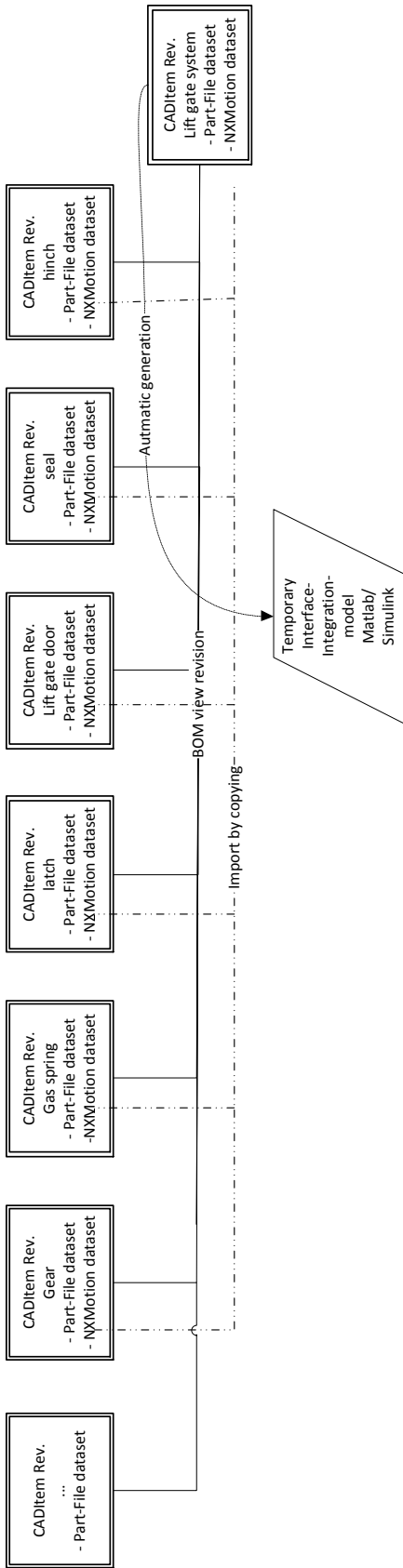


Figure 5 33 Temporary Interface Integration Model for Matlab/Simulink

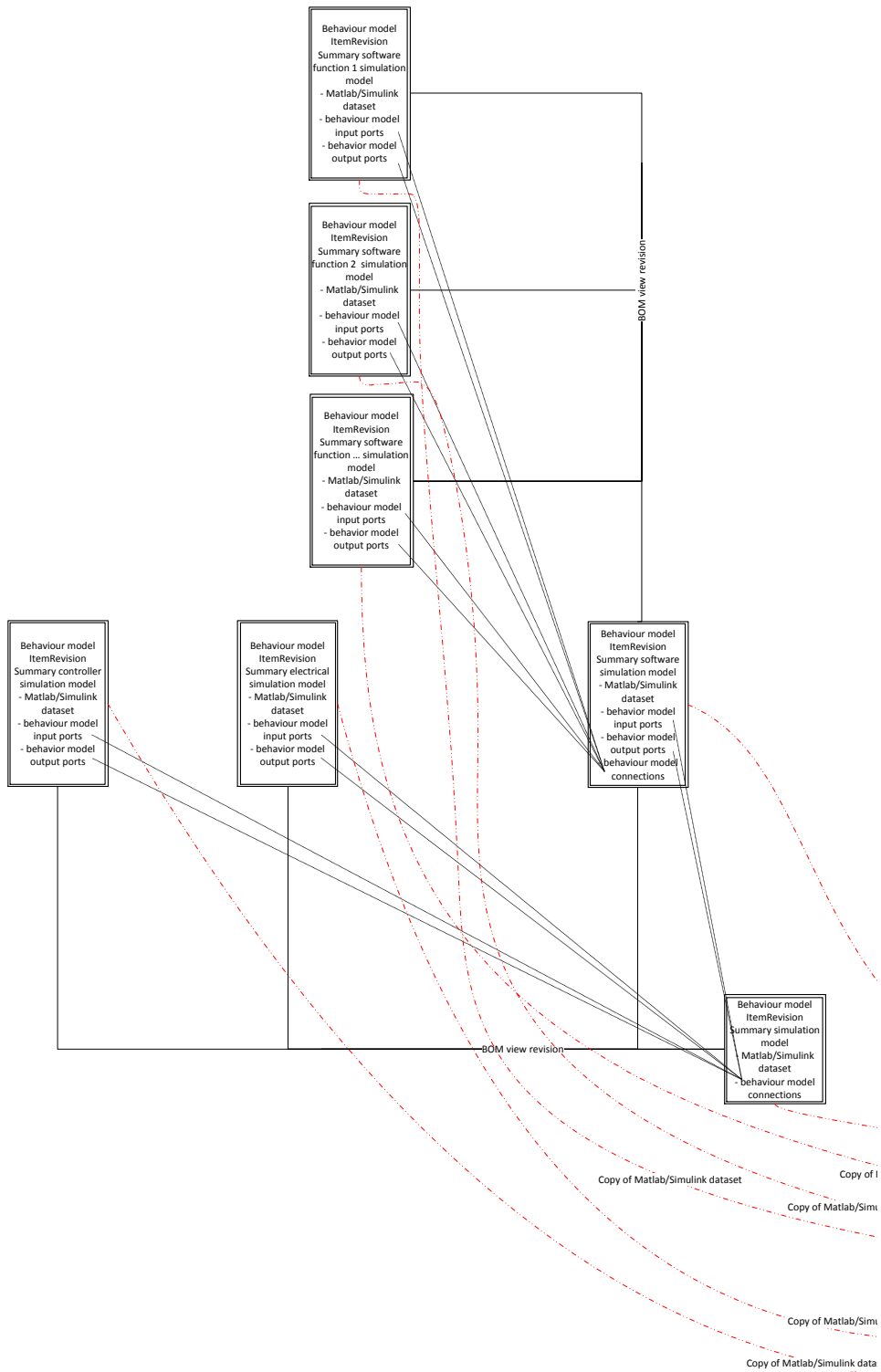


Figure 5 34 Summary Data Management of Case 3



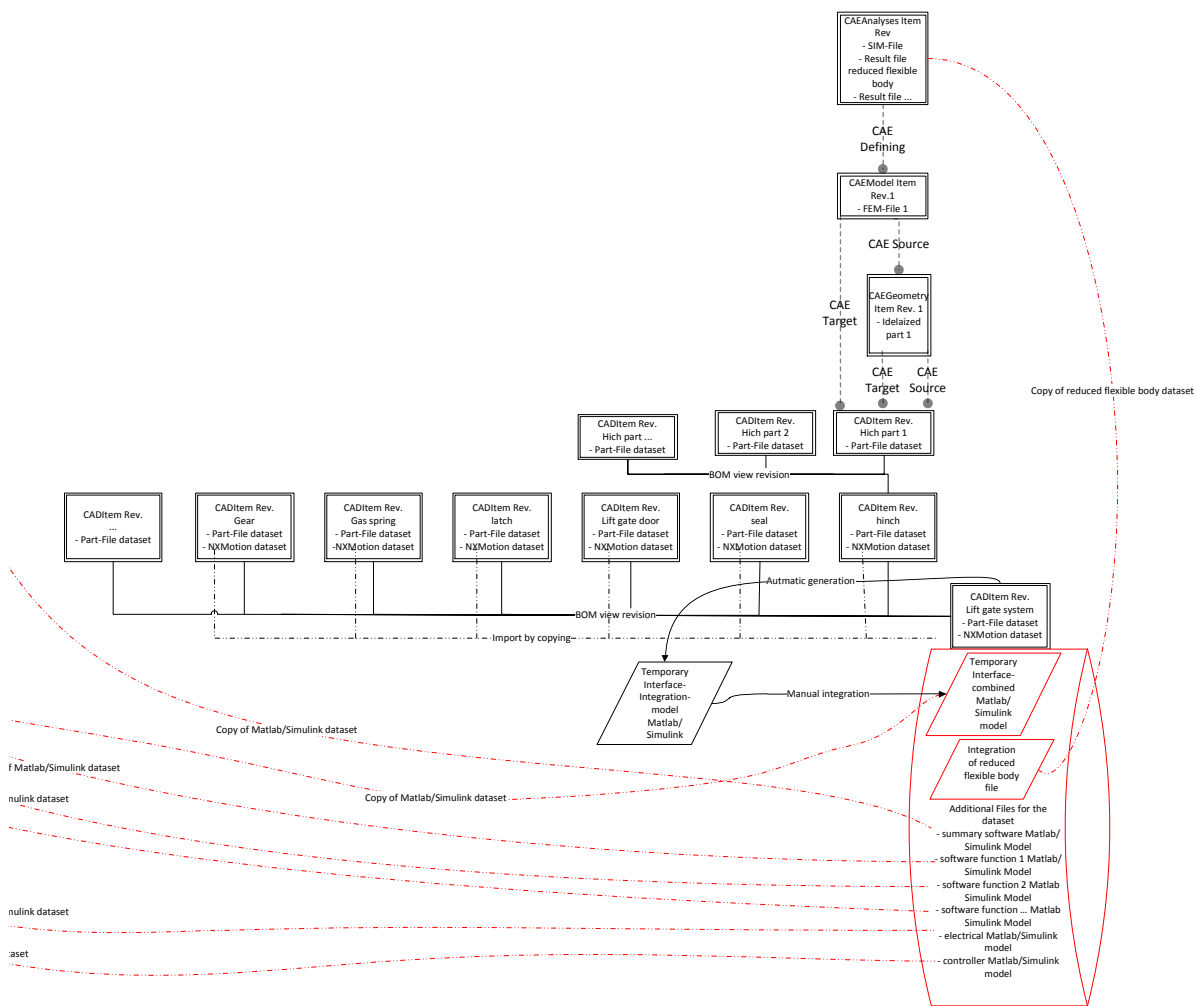


Figure 5 34 Summary Data Management of Case 3

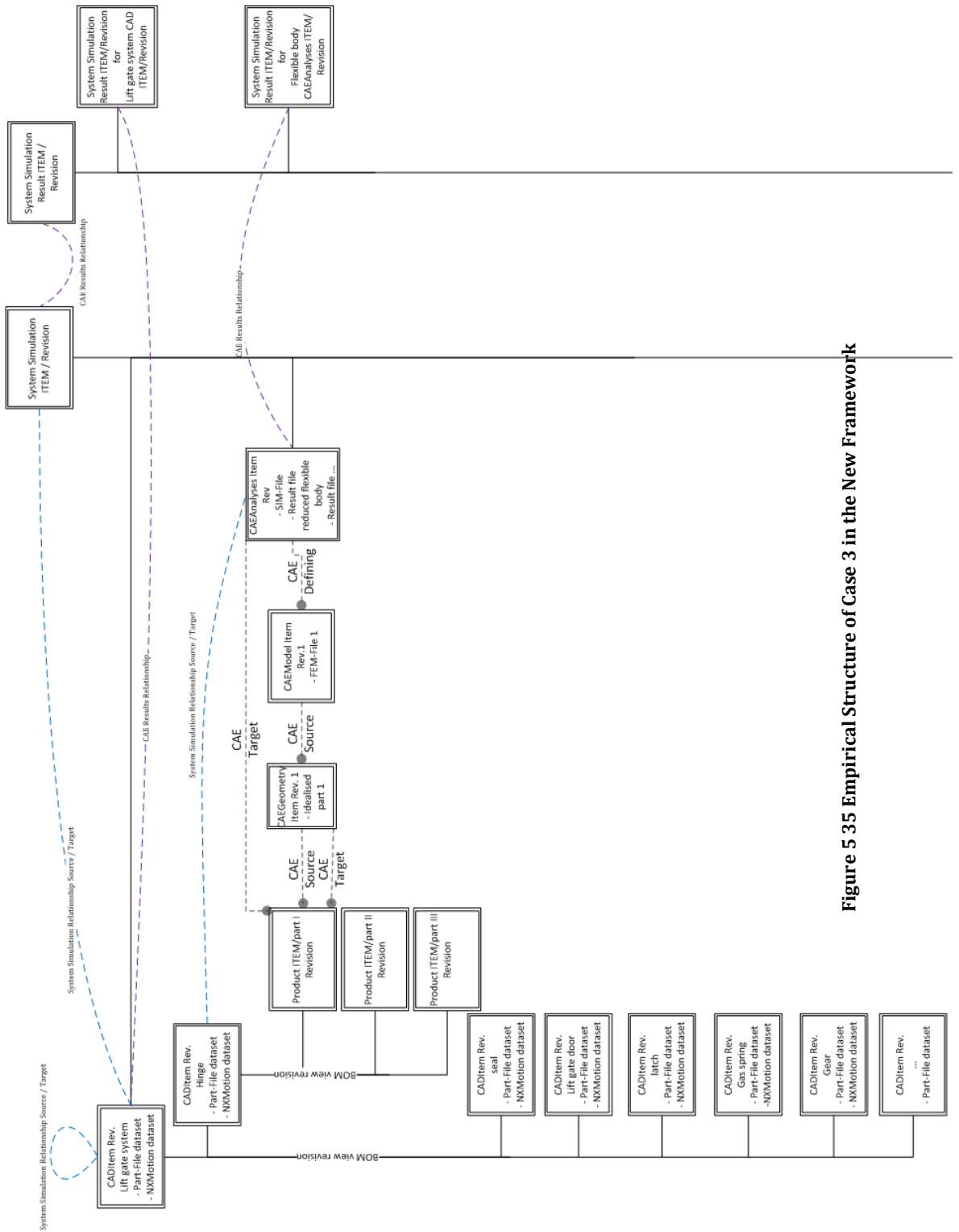


Figure 5 35 Empirical Structure of Case 3 in the New Framework

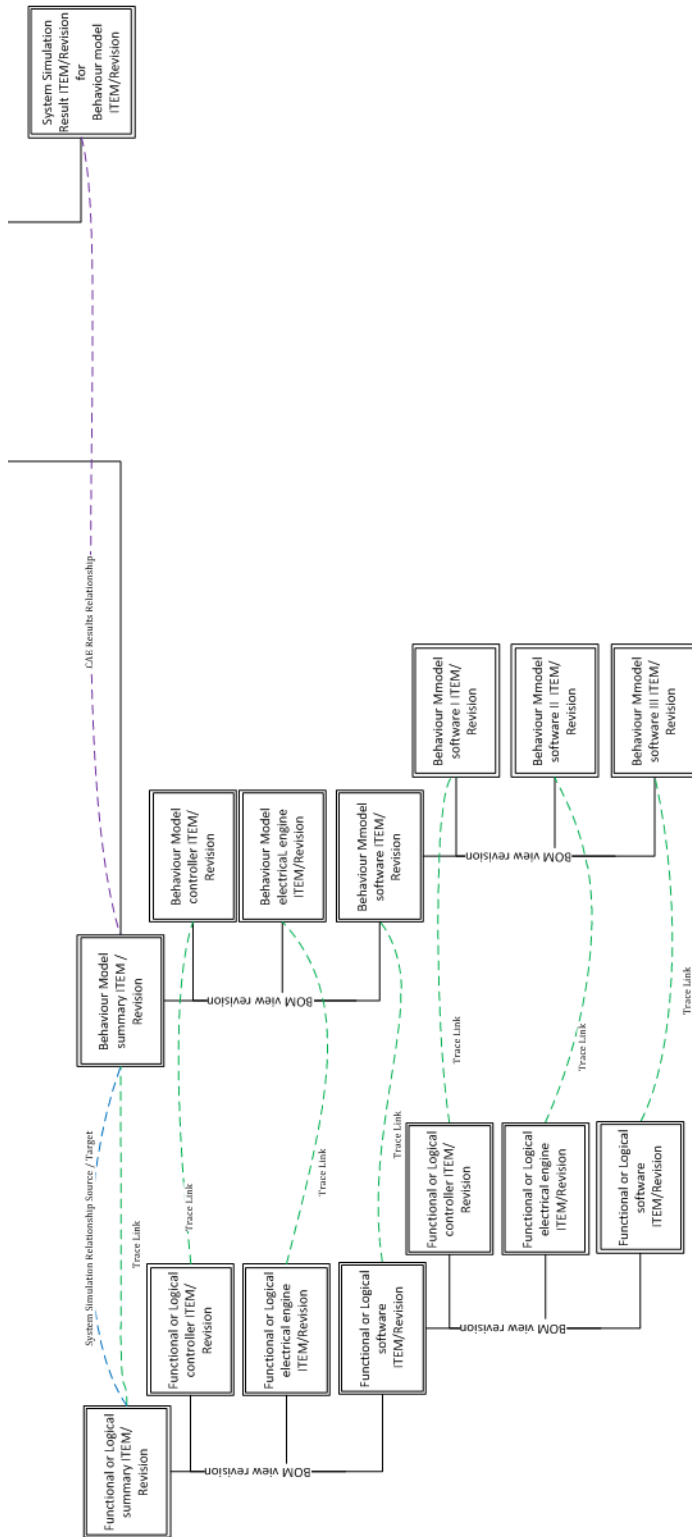


Figure 5 35 Empirical Structure of Case 3 in the New Framework

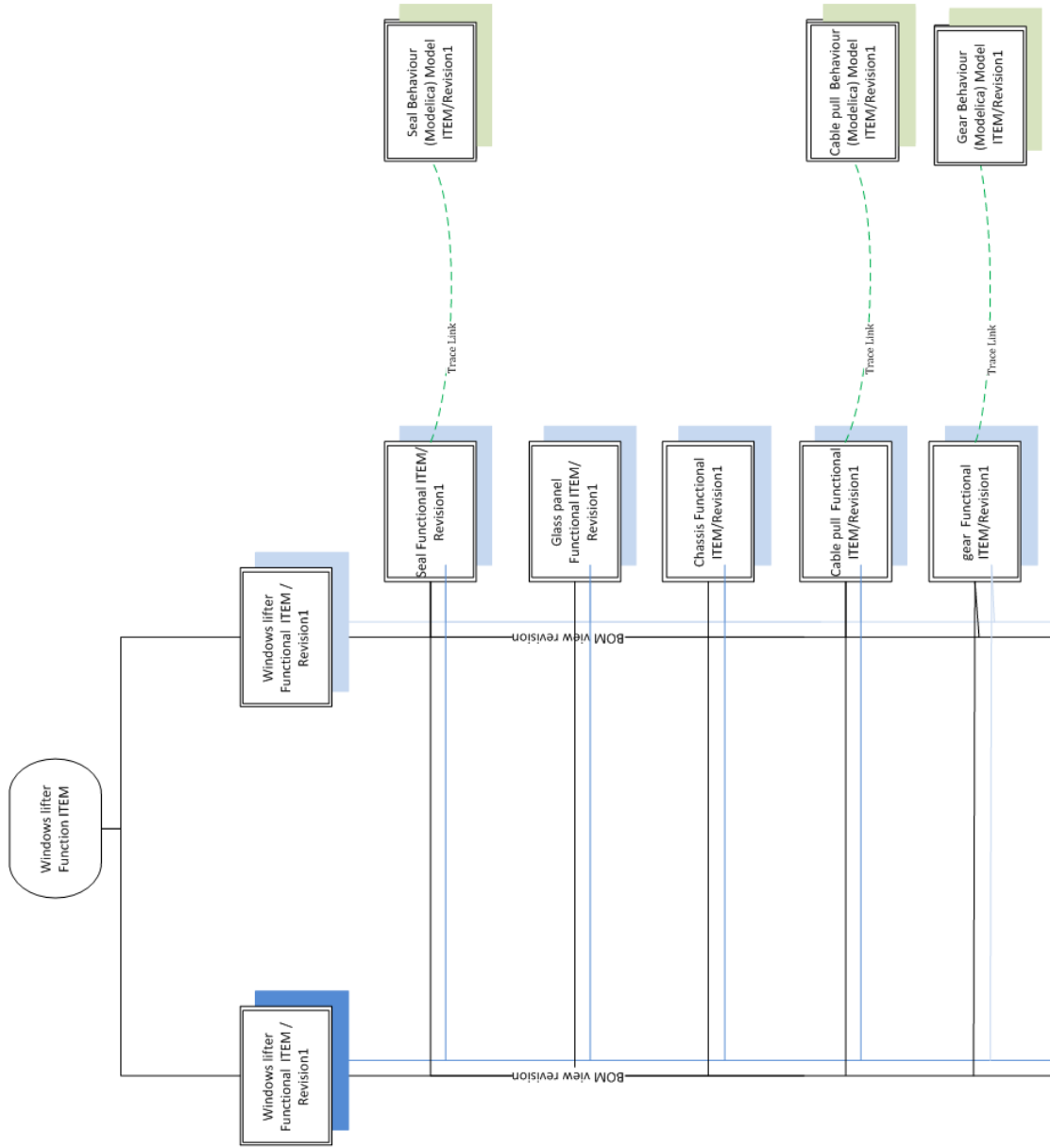


Figure 5 51 Functions and Behaviour Models during the Development Process

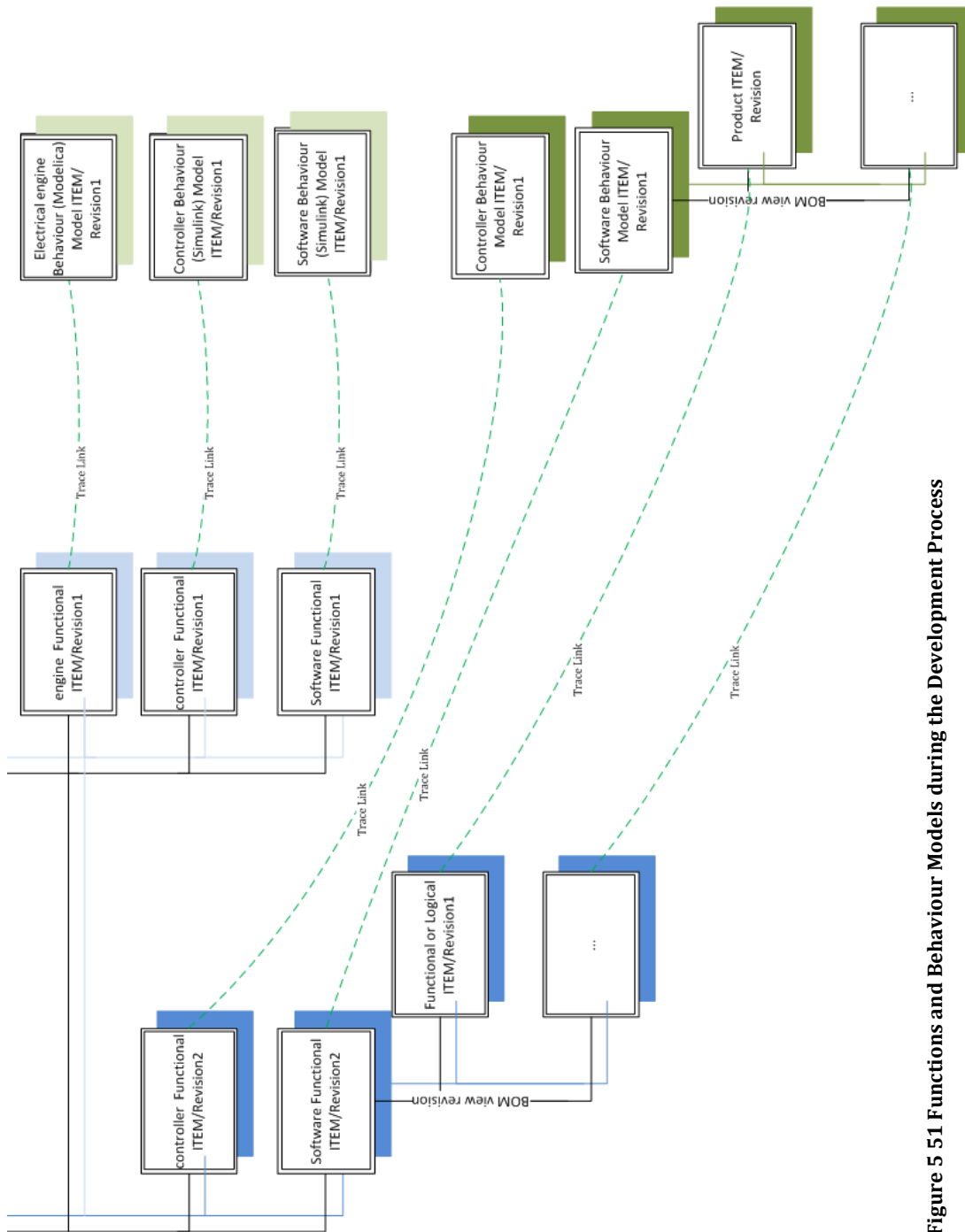


Figure 5 51 Functions and Behaviour Models during the Development Process

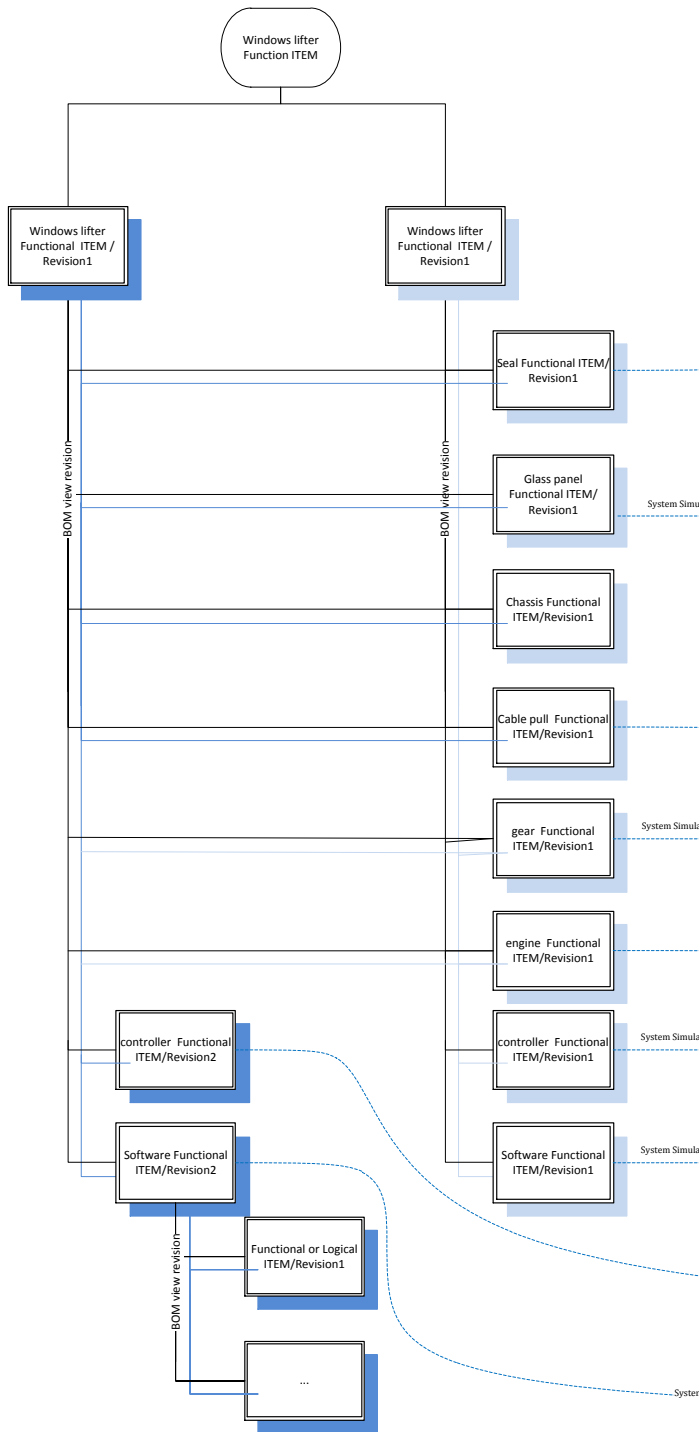


Figure 5 53 Simulation Model Traceability of Represented and Sourced System

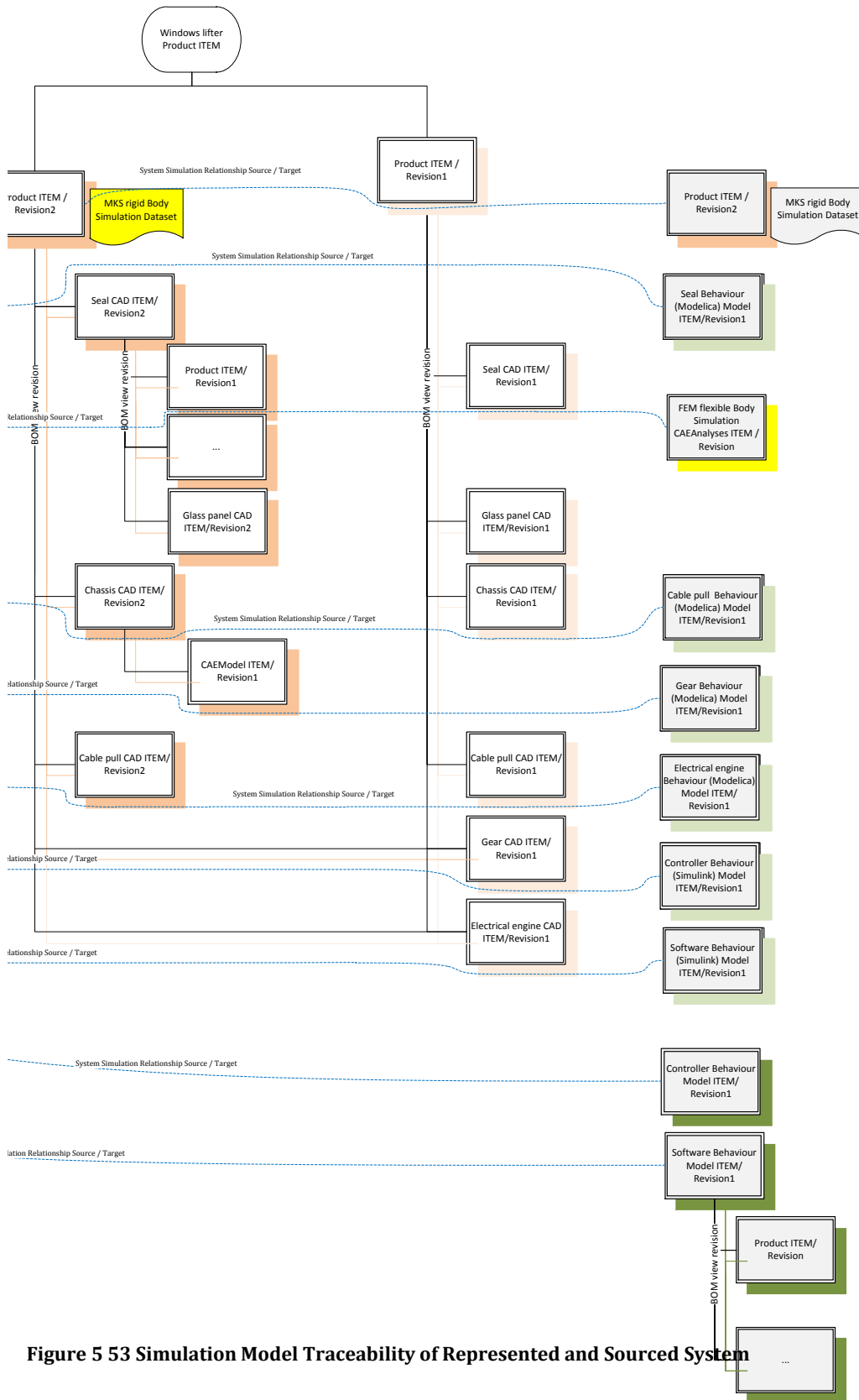


Figure 5 53 Simulation Model Traceability of Represented and Sourced System

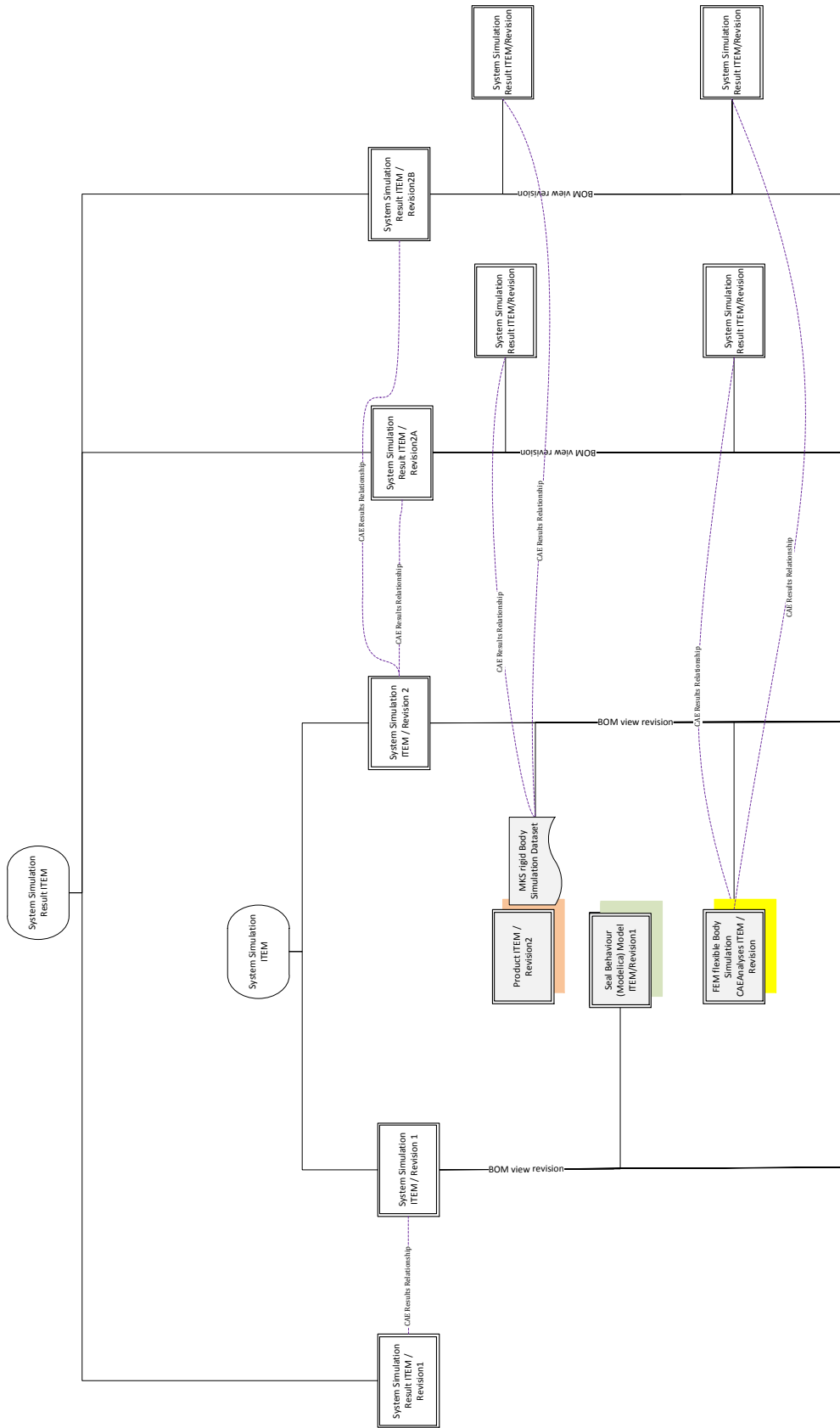


Figure 5 54 System-Simulation and Result Management



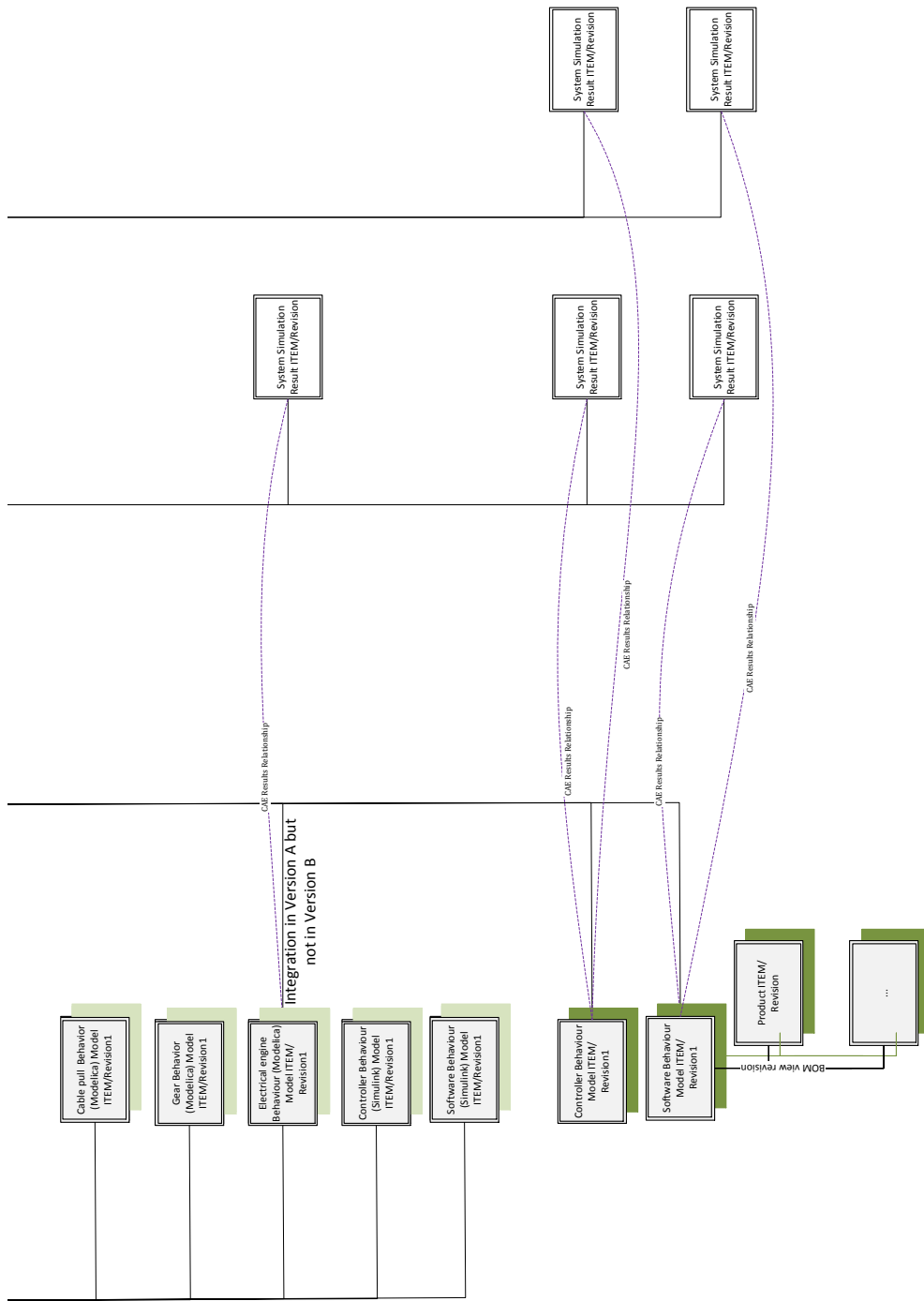


Figure 5 54 System-Simulation and Result Management

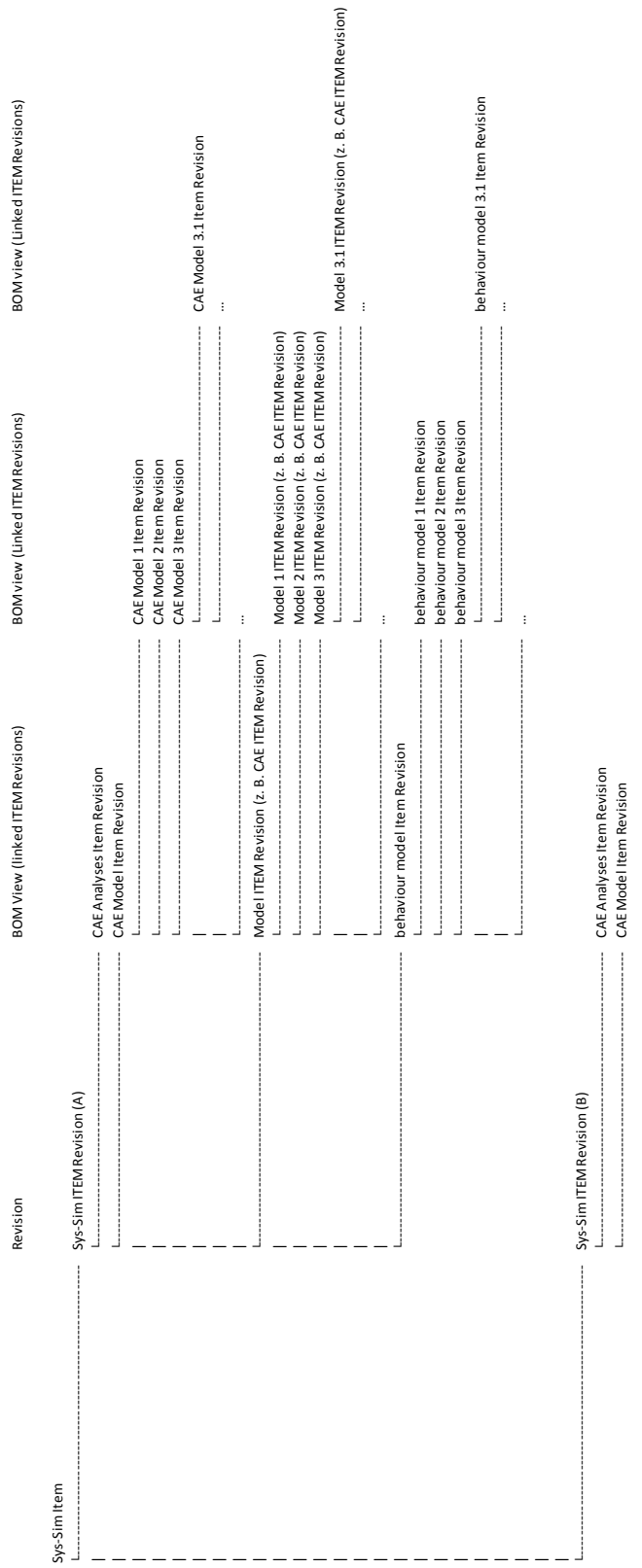


Figure 6 6 Ordering of the System-Simulation Structure

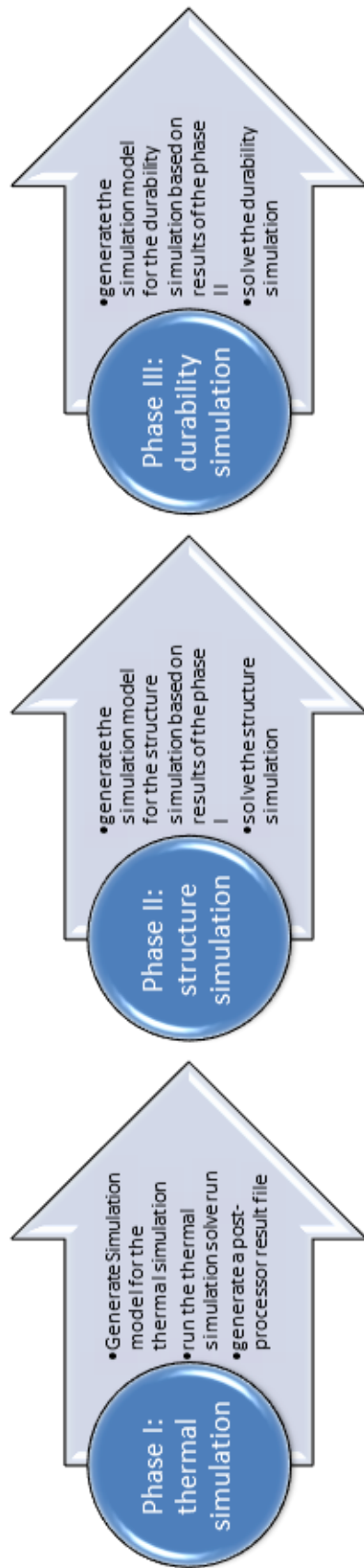


Figure 6 7 Serial Simulation Process of Case 1

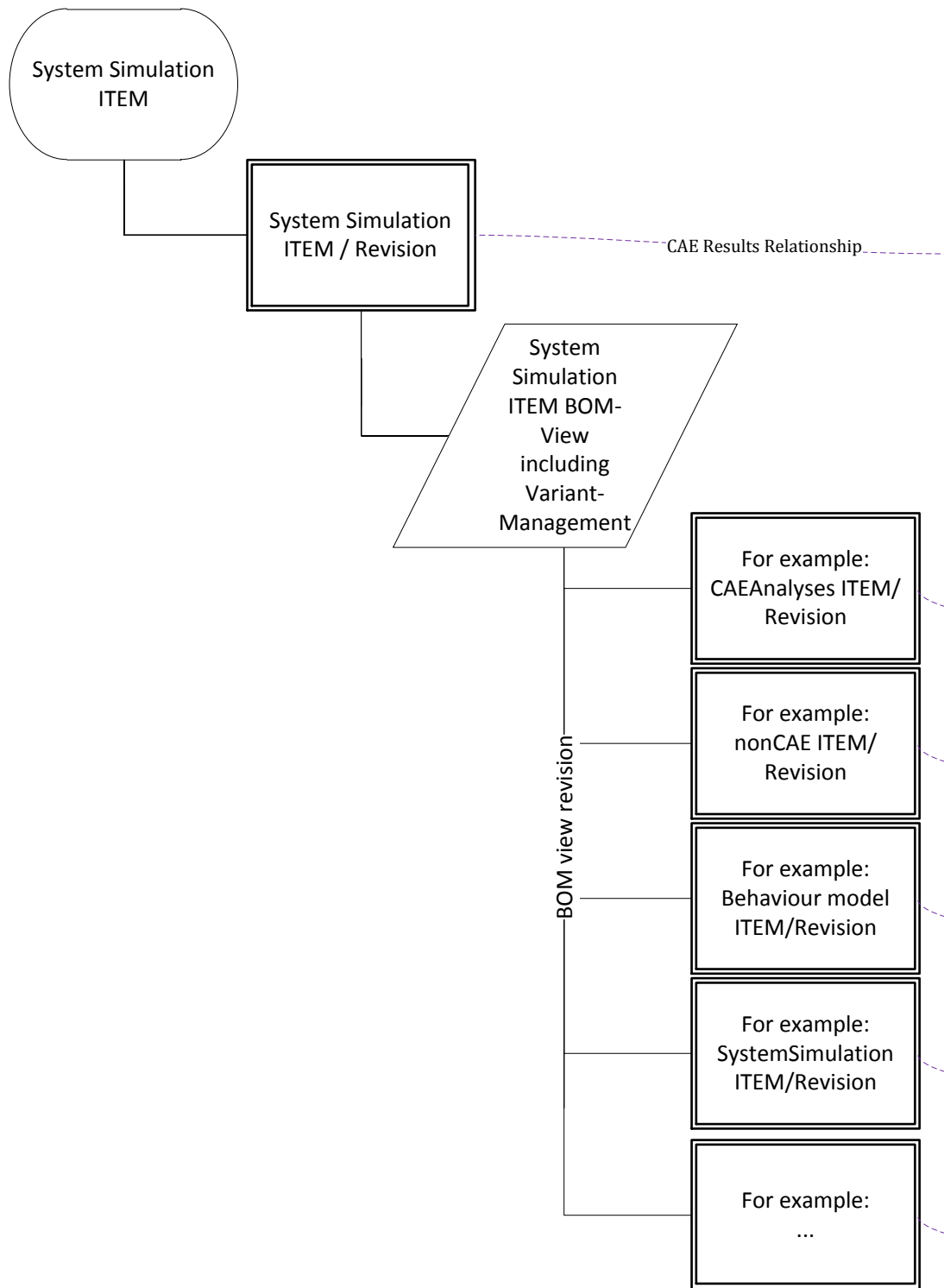


Figure 6 24 Ordering Rules for Sys-Sim ITEM Revision and Sys-Sim Result ITEM Revision

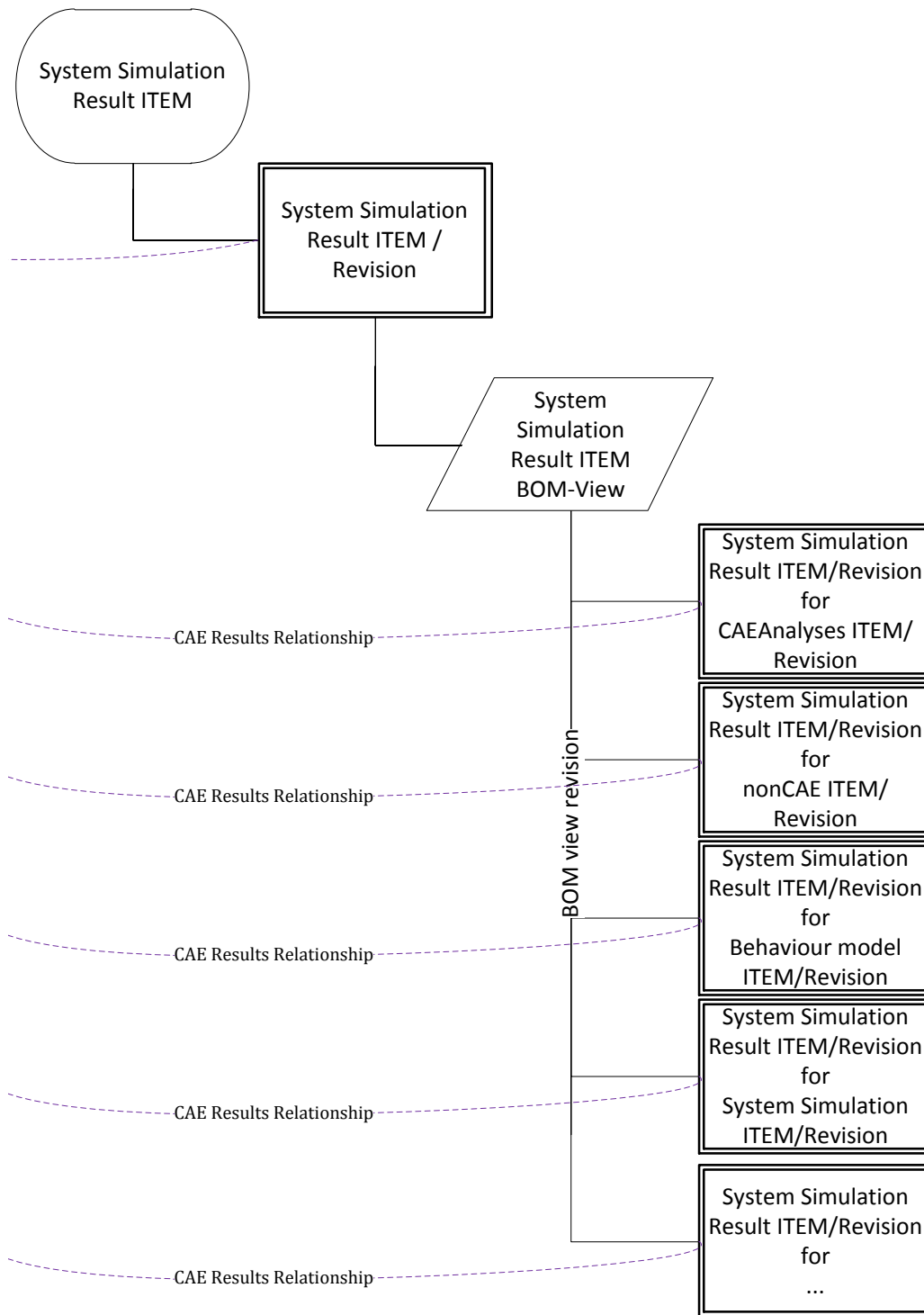
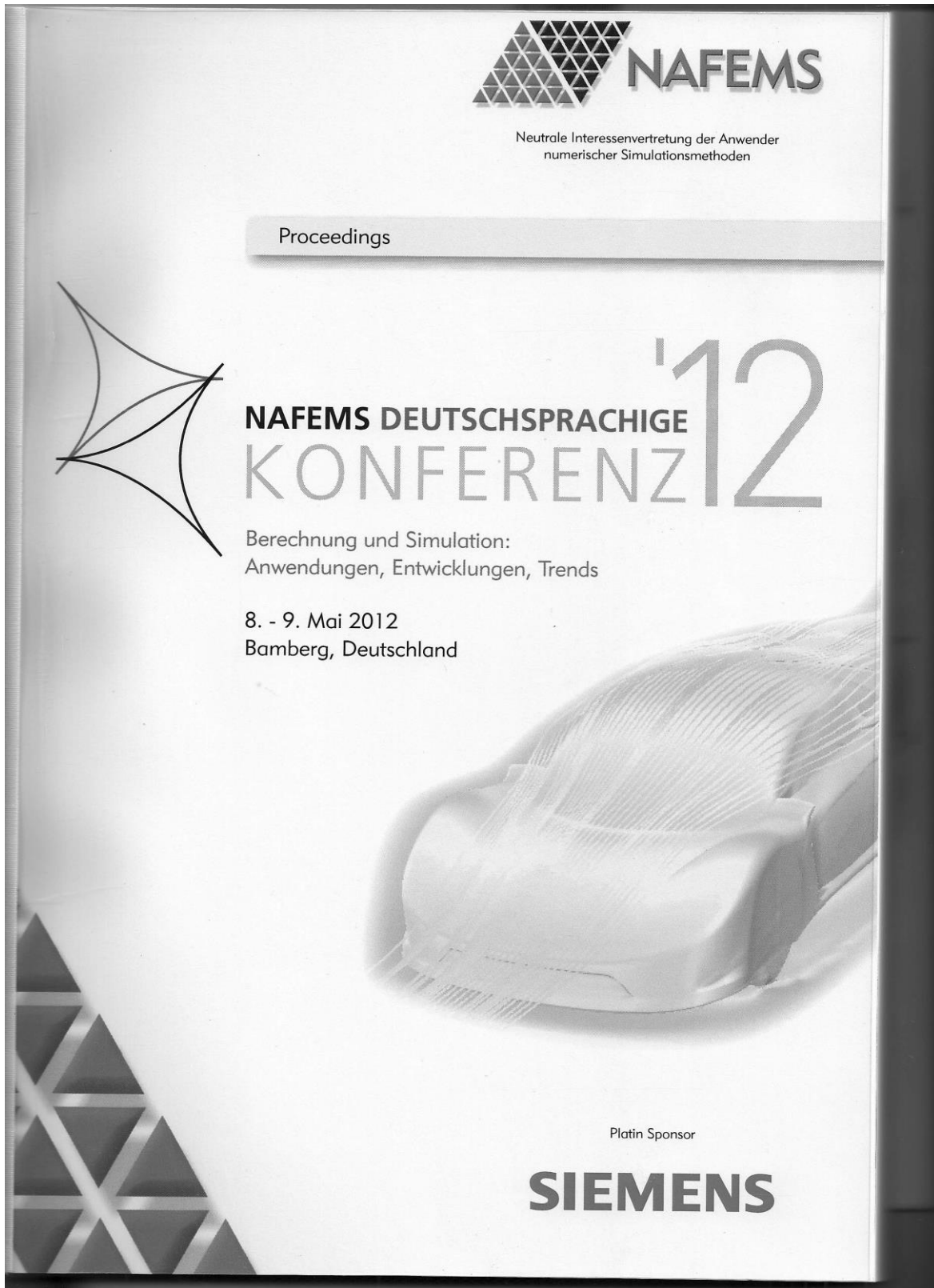


Figure 6 24 Ordering Rules for Sys-Sim ITEM Revision and Sys-Sim Result ITEM Revision

## 9.4 APPENDIX D PUBLISHED PAPER



## Framework for System-simulation based system engineering development methodology

Dipl. Ing. (BA) Michael Mahler (B. Sc.)

Siemens Industry Software GmbH, Germany

### 1 Abstract

The development of products and their systems with high complexity among the subsystems and the controllers is heavyset and extensive. Minor changes and small numbers of missing information could cause the fail of the product in a later state. To optimize the situation the development methodology should be improved. The idea: Controlling the development process will help improving the quality of the development phases. This should be done by controlling the maturity at the different development phases. Described will be the theory and a framework for realization.

Die Entwicklung von Produkten und Systemen mit hoher Komplexität zwischen den Teilsystemen und dem Kontroller ist sehr schwierig und aufwendig. Kleine Änderungen und minimal fehlende Informationen können zum Verfehlen des Entwicklungsziels führen. Um diese Situation zu verbessern soll die Entwicklungs-Prozess-Kette verbessert werden. Die Idee: Den Entwicklungs-Prozess zu kontrollieren sollte helfen die Qualität der Entwicklung in den Entwicklungs-Schritten zu verbessern. Dies sollte durch die Kontrolle des Reifegrads in den Entwicklungs-Abschnitten erreicht werden. Beschrieben wird die Theorie wie auch eine Umgebung für deren Realisation.

### 2 Improvement of the development-process-methodology

Existing development methodologies should be improved to generate a better support to develop complex systems. High influence and complexity of subsystems causes high subsystem-controller-dependencies. Minor changes and small numbers of missing information could cause the fail of the product in a later state, specially in the development of the controller.

#### 2.1 State of the technology

The most conventional product-development-methodology will be described in (Pahl & Beitz, 1993). This product-development-methodology is based on a serial process: product idea, product planning, product construction. Product construction includes concept development - constructive concept - constructive development, prototype production, testing. In case of mechatronics systems this serial process was improved to the model-based mechatronics development methodology. Based on a design-loop an optimization of the mechatronics concept will be done at an early stage of the development process (Lennon, November 21, 2007) (Klotzbach, Oedekoven, & Grassmann). Therefore the product-construction-phase will be exchanged or better improved by the loop of mechatronics-concept-phase including model generation, parameter identification, analyses and syntheses. But there is no system responsible to take care about system functions from the starting point of the development till the end. A cross-function was built up called system engineering concentrating on a system viewpoint. This authority will generate and check product requirements. It will take care about the functionality of the system in each product development level (Boucher, October 2009) (Stark, Beier, Wähler, & Figge, 2010).

"The system engineering process is a comprehensive, iterative and recursive problem solving process which is suitable for the development of mechatronics products. ... First step is the analysis of customer requirements in order to derive functional and performance requirements. Subsequently, functions are identified, decomposed and allocated to the requirements. During synthesis the product



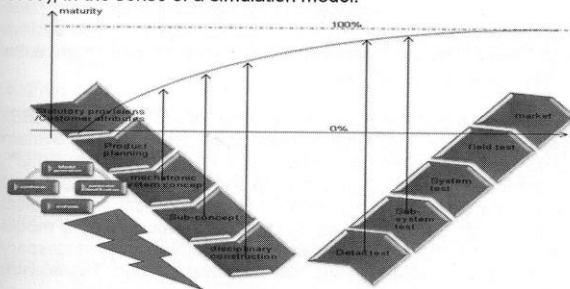
is defined in terms of physical and software elements and afterwards verified against the requirements..." (Stark, Beier, Wähler, & Figge, 2010)

The model-based mechatronics development methodology and system engineering will get unified to model-based system engineering (Eigner, Gilz, Hollerith, & Zafirov, 07. Nov. 2011). The model idea of model-based mechatronics development methodology (like CAD models or executable models like equation based models) will help to verify the requirements generated in the system engineering development methodology. This check of the requirements against an early development stage should identify the failing or bad functions parallel to the development-process.

## 2.2 Controlling the development process by the maturity

The mechatronics system concept will be worked by using simulation to optimize the concept. But in the reality systems with high influence and complexity of the subsystems will be at risk to fail. An improvement of the development methodology will be useful to optimize the development of systems as described before. Those systems will have most of the time high subsystem-controller dependencies (Dr.-Ing. J., Dipl.-Ing. S., & Prof. Dr.-Ing. H., 2010), where small tolerances in the subsystems could causes failure of the system. To identify a failure of a system in the case of tolerance scenarios (maximum, minimum, ...) will not be integrated in the system-engineering-process. Therefore the idea is to control the development process by the maturity. This should identify a failure of the system in an early state and reduce failing in later states and keep the development in time.

The maturity will be calculated as the difference of the ready-product-value to the simulated system-value. We presume that the maturity will be ideal over the development in a tangent hyperbolic curve of the first quadrant in a Cartesian coordinate system. To identify the maturity measurable values have to be worked out like velocity, force, current and so on. As a base for generating the necessary data, (Lennon, November 21, 2007) the model has to be an executable model (Jeremy, 20th - 23rd April 2009), in the sense of a simulation model.



## 3 Framework for realization

A framework supporting the development process controlled by maturity has to integrate multiple disciplines. These disciplines could be split in two categories: first the data and process management including the stored product describing data and second the system-simulation of the product. These two categories have to be married and doable along the product development process.

Testing could be used instead of system-simulation. But there are two problems influencing the usability of testing:

- a prototype is producible in a late state of the development process
- the quality of the prototype will not represent the tolerance field.

For the representation of the tolerance field the testing has to be increased to a field test.

In the case of systems with high influence of tolerances on the controller the system failing could be identified at the field test. What will cause extreme costs because the development has to loop extreme backward and the time to market will be out of plan.

### 3.1 System-Simulation of the product

In the development process we have to decide if we like to describe our system-simulation based on existing product describing data like CAD geometry or equation based product descriptions like Newton axioms. In an early stage of the development process product describing data are not available and the only choice would be to describe the system with equation based descriptions. Equations are independent from the disciplines. That means the equation based simulation (EBS) representation of the product could include mechanical, electrical, software, etc.. representations in one executable model.

Later when the first models are built up, this begins with a skeleton model/sketch/draft, a decision have to be made between a model-based or on equations-based simulation. These decisions have to be repeated for all model improvements. In the case that the product should fully or partly be represented by a model-based simulation, EBS will no longer be used. Instead of those, simulation tools for finite element method (FEM) or multi body simulation (MBS) or computer fluid dynamics (CFD) will be used. In that state the physical disciplines described above in the EBS declaration is now changing to a mathematical simulation tool discipline. In the case to represent the product by multiple simulation tools the simulation tools have to be coupled with each other, while the individual simulation tools have to be possible to cooperate as a co-simulation.

A framework for this is NX. NX includes huge functionalities for FEM, MBS as well CFD and has an interface to EBS like Matlab/Simulink too. NX also supports different solvers like Nastran, Ansys, Recurdyne, Adams, ...

NX supports the associativity of the simulation models to the CAD models by overwriting parameters only for the simulation (not for CAD) parameters to represent the tolerance maximum or minimum values.

### 3.2 Data and process management

Essentially the product data management (PDM) and product lifecycle management (PLM) tools have to fulfill all requirements coming from the conventional development methodology and also system engineering should be supported. With Teamcenter and Teamcenter Foundation as well Teamcenter System Engineering & Requirements Management this will be achieved.

Additional to that the PDM/PLM tool should be able to integrate the equation based simulation (EBS) models in a system engineering environment. The EBS-models have to be embedded in System Engineering because in System Engineering the requirement – functional – logical – physical (RFLP) structure will be represented. The EBS-models could belong to the functional or logical or physical level. But the functional model could include additional functions the same for logical and physical models. The representation of these in a single EBS model file is not possible, multiple EBS model files have to be loadable and therefore the PDM/PLM system should include the necessary information about these. In Teamcenter these knowledge will be available through Teamcenter System Engineering & Requirements Management and Mechatronics Process Management.

The simulation models based on models (not executable) will be built up, these model should be linked with the model base. The creation of these model-based simulation models will need multiple steps like model-based idealization for the simulation model, meshing and so on. These structures should be traceable. Teamcenter will support these with Teamcenter Simulation Process Management.

## 4 Summery

To build up a framework for system simulation-based system engineering development methodology needs careful work. Different tools have to be used and linked to each other. These tools have to be selected thoroughly to fulfill all the requirements on it. Requirements are coming from the IT to create a continuous and integrated IT system and requirements are coming from the operating department to solve their problems.

Today the use of one software tool is not possible, multiple software tools have to be integrated. These software tools could be categorized in simulation tools as well in data and process management tools.

The data and process management tools should collect and organize all data and processes of the development methodology including models for



- Requirements
- Functions
- Logical
- Physical
- CAD
- Simulation models (executable)

The simulation tools should be model-based or equation-based. A decision comes up when the change from equation to model-based simulation have to be done. The decision could be to go on partially with a model based simulation, for example to simulate rigid bodies with MKS and connect this with a simulation of the controller unit represented by EBS. There for the simulation tools have to be linkable.

Simulation systems and tools will produce knowledge about the simulated object. The data and process management tool will store the knowledge and make it popular or push it to persons how and software tools which will need it. The reduction of missing possibility for identify knowledge in a company will improve the quality of the development stages. Organizing the knowledge and controlling the product parallel to the development process will help to stay on time, on budget and on target cost (Krauss, Becker, Paulukuhn, Weber, & Schröder, *Forschung und Entwicklung managen - Erfolgsfaktoren im Maschinen- und Anlagenbau*, 2005) (Krauss, Becker, Paulukuhn, Weber, & Schröder, *Forschung und Entwicklung managen - Erfolgsfaktoren in der Automobilindustrie*, 2005).

## 5 Works Cited

- Boucher, M. (October 2009). *System Engineering Top Four Design Tips to Increase Profit Margins for Mechatronics and Smart Products*. Aberdeen Group.
- Defense, D. o. (2001). *Systems Engineering Fundamentals*. Virginia: Defense Acquisition University Press.
- Dr.-Ing. J., F., Dipl.-Ing. S., G., & Prof. Dr.-Ing. H., B. (2010). Ganzheitliche Simulation von mechatronischen Schwingungstilgern für automobiler Anwendungen. *NAFEMS Magazin* 2/2010.
- Eigner, M., Gilz, T., Hollerith, T., & Zafirov, R. (07. Nov. 2011). Methoden für die modellbasierte interdisziplinäre Produktentwicklung. 3. *PLM Future Workshop* (S. 26 slides). TU Kaiserslautern: VPE Lehrstuhl für Virtuelle Produktentwicklung.
- Jeremy, G. (20th - 23rd April 2009). Linking Architecture Models and Synthetic Environments - An Integrated Project Support Environment. Loughborough University: 7th Annual Conference on Systems Engineering Research 2009.
- Klotzbach, S., Oedekoven, S., & Grassmann, O. *Optimierung im mechatronischen Entwicklungsprozess*. Köln: DMECS Development of Mechatronic Systems.
- Krauss, H.-D., Becker, D., Paulukuhn, D. L., Weber, D. P., & Schröder, D. J. (2005). *Forschung und Entwicklung managen - Erfolgsfaktoren im Maschinen- und Anlagenbau*. KPMG Deutsch Treuhand-Gesellschaft Aktiengesellschaft Wirtschaftsprüfungsgesellschaft, deutsches Mitgliedsunternehmen von KPMG International, einer Genossenschaft schweizerischen Rechts.
- Krauss, H.-D., Becker, D., Paulukuhn, D. L., Weber, D. P., & Schröder, D. J. (2005). *Forschung und Entwicklung managen - Erfolgsfaktoren in der Automobilindustrie*. KPMG Deutsch Treuhand-Gesellschaft Aktiengesellschaft Wirtschaftsprüfungsgesellschaft, deutsches Mitgliedsunternehmen von KPMG International, einer Genossenschaft schweizerischen Rechts.
- Lennon, T. (November 21, 2007). Model-based design for MECHATRONIC SYSTEMS. *Machine Design*, 42-51.
- Pahl, G., & Beitz, W. (1993). *Konstruktionslehre - Methoden und Anwendung*. Berlin: Springer Verlag.
- Rauchenberger, J., Spiegelberger, B., Stetter, R., & Schmitt, R. (2009). Professionalisierung des mechatronischen Entwicklungsprozesses durch Anwendung eines Reifegradmodells / Professionalizing the Mechatronical Development Process by the Application of a Maturity Model. *Industrie Management* 25, 53 - 56.
- Schwabe, S. (2009). Modellbasierter Systems-Engineering-Prozess. *ECONOMIC ENGINEERING* (03), 58.
- Stark, R., Beier, G., Wähler, T., & Figge, A. (2010). *Cross-Domain Dependency Modelling - How to achieve consistent System Models with Tool Support*. Fraunhofer Institute for Production Systems and Design Technology; Berlin as well Technische Universität Berlin. INCOSE.



In cooperation with



MULTIPHYSICS  
The International Society of Multiphysics

NAFEMS European Conference  
**MULTIPHYSICS  
SIMULATION 2012**  
16 - 17 October 2012 Frankfurt | Germany

Gold Sponsor



Silver Sponsors





## Multiphysics Simulation of Molding Tools for Qualified Designers and Engineers

Roland Schmidt (SCHMIDT Gesellschaft für Werkzeug- und Formentechnik mbH);  
Dipl. Ing. (BA) Michael Mahler (B. Sc.) (Siemens Industry Software GmbH)

### 1 Abstract

Molding tools will be used in a wide range in the industry. Nevertheless, the molding tools are expansive to be developed and produced. To manufacture competitive plastic parts economically the forming errors have to be reduced and the forming speed has to be increased (1). Therefore the molding tools should be improved. But most of these improvements will cause increasing forming errors. The higher applied loads will cause deformation of the forming faces as well as damage of the molding tools. Simulation should forecast the attitude and the durability of molding tools as well forming errors. These simulations have to including applied mechanical as well thermal loads. These is necessary additional to the injection simulation.

### 2 Introduction

Molding tools are used in a wide range throughout the industry. "Injection molding is an ideal process for fabricating large numbers of geometrically complex parts. Many daily used items are injection molded: mobile phone housings, automobile bumpers, television cabinets, compact discs and lunch boxes are all examples of injection molded parts." (2) However, the molding tools are expansive to be developed and produced. For reducing the work piece costs a high lot size has to be realized. These will end in the interest to boost the lot size per time. But the quality of the work piece has to be guaranteed. In summary: In order to manufacture competitive plastic parts economically the forming errors have to be reduced and the forming speed has to be increased (1). These affect the forming process.

### 3 Forming process

"Injection molding is a cyclic process. Initially, the mold is closed to form the cavity into which the material is injected. The screw then moves forward as a piston, forcing molten material ahead of it into the cavity. ...When filling is complete, pressure is maintained on the melt and the packing phase begins. The purpose of the packing phase is to add further material to compensate for shrinkage of material as it cools in the cavity. At some time during packing, the gate freezes and the cavity is effectively isolated from the pressure applied by the melt in the barrel. This marks the beginning of the cooling phase in which the material continues to cool until the component has sufficient mechanical stiffness to be ejected from the mold. ... When the molded part is sufficiently solid, the mold opens and the part is ejected. The mold then closes and the cycle begins again." (2)



### 4 Improvement of the forming process

To improvement the forming speed following has to be improved and optimized:

1. In the modeling closing phase the parts will be moving. To speed up the closing process the mass of the parts should be reduced and the drive force increased. These causes applied loads resulting stress and deformation.
2. The Injection could be done with a higher injection velocity and higher pressure as well a higher injection temperature (maybe caused by improved injection materials) (3). Most forms are idealized for a better mold material behavior in the injection process. The injection pressure and mold material temperature will cause applied loads resulting in stress and deformation.

3. The pressure during the packing phase will be increased (3). These should reduce the errors due to material shrinkage. This pressure will cause higher applied loads resulting in stresses and deformation.
4. The cooling process could be done faster and in a controlled manner (4). Therefore cooling-pipes will be introduced in the molding tools. This cooling will cause higher applied loads resulting in stresses and deformation.
5. The Ejection process could be improved on the same way as the mold closing process in phase 1. The mass of the parts should be reduced and the drive force increased. These causes applied loads resulting stress and deformation.

Speeding up the process by reducing the mass of the molding tool and increasing the loads will result in increasing forming errors and lower life of the tool. A proper optimization of process parameters and tool sizing are necessary. This will be done with a coupled physics analysis by taking into account all mechanical and thermal loads during a complete molding cycle.

### 5 Simulation of the molding tool

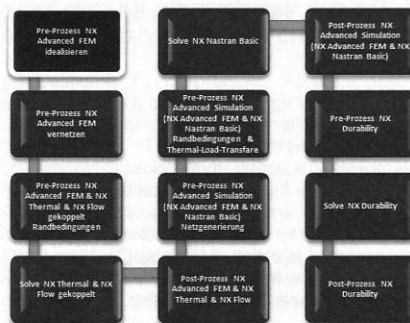
The simulation of the modeling tool should integrate the following physical simulation disciplines:

1. Co-simulation of thermal and flow
  - a. Flow simulation of the coolant
  - b. Thermal simulation of the coolant flow and the molding tool → conjugate heat transfer simulation
2. Calculating deformation and stress of the molding tool caused by all applied mechanical and thermal loads
3. Assessing the damage of the structure due to all applied mechanical and thermal loads.

#### Phase 1:

The flow of the coolant is necessary to get the right cooling caused by the coolant. But we have to couple the flow of the coolant with the thermal simulation of the coolant realized with the NX CAE solutions: NX Flow and NX Thermal (5). The fluid volume of the coolant channels was modeled from the solid model of the molding tool with NX synchronous (6). Both geometries, the flow as well as the modeling tool has to be meshed. The NX mesher could check the mesh quality direct and re-mesh till the predefined mesh quality is achieved (7). The flow mesh hast to be coupled to the solid mesh for the thermal simulation. NX provides therefore an easy surface to surface coupling without synchronizing the meshes (8). The applied loads and boundaries as well as solver parameters are:

- The temperature of the injection mold at the contact faces to the injection material
- The free convection to the environment
- The coolant will be described between the inlet and outlet as Recirculation Loop with a predefinition of flow velocity and the heat exchanger parameters like heat transfer coefficient, convection area and temperature. (coupling 3D and 1D flow simulation)
- A mixing length turbulence model has been chosen
- Simulation results will be reached in the case of Steady State. The stress difference between the warmest and coldest temperature in one production cycle process will be seen as negligible



#### Phase 2:

For the structural analysis only the solid mesh of the molding tool is required. The physical setup from phase 1 is cloned and directly can be used as an initial setup for a structural simulation. The mesh of the fluid volume and the connected setup are deleted easily. In order to complete the physical setup of the structural analysis bolt connections are defined with a dedicated toolset, pressure loads from the mold injection, closing forces and bolt preload forces, as well the boundary conditions are added. Contact face pairs are detected automatically. In most cases the contacts are defined as surface-to-surface contacts with friction. Contacts are necessary to accurately



take into account the movement in the contact areas and the contact pressures. The temperature field of the thermal analysis is used as initial condition for the NX Nastran solution (8) using the linear solver SOL 101. Two subcases have to be defined,

- subcase 1 stress of the warmed up molding tool loads: Thermal loads representing the steady state thermal loading of the working molding tool → reached once a
- subcase 2 stress of a lot production: applied loads of one production cycle (closing force, injection pressure, ...) → reach for each work piece

Phase 3:

The required mesh quality for fatigue calculation has to be checked and generated in front of the fatigue calculation. NX Advanced Simulation therefore provides an automatic mesh refinement algorithm taking into account local stress error deviations (7). This results in a refined mesh in areas with high deformation energy. The fatigue load cases should represent a production day. There for the load cases will be summarized like following:

- 1 cycle of the subcase 1 representing the loads of a warmed up molding tool
- {Number of produced work pieces a day} cycles of the subcase 2 representing the loads of a lot production

To optimization the molding tool only relative and no absolute fatigue results are necessary. However, it is common practice to relatively compare fatigue results in order to compare different design variants. These practice will be used to improve the molding tool. Nevertheless, NX Advanced Simulation offers both capabilities, calculating absolute and relative fatigue predictions.

## 6 Summary

The goal to improve the molding tool will be reached. Cooling as well as mechanical loads could be introduced to this view. A good interpretation of different design variants as well as design optimization will be archived. The user shouldn't be high end user. This simulation can be performed by qualified designers and engineers. Caused by

- the easy usability of the software,
- the integration of CAD modeling functionalities into the Pre-Process,
- the automatic mesh generating and quality check,
- the intuitive use of the function windows for applying loads and boundary conditions
- and the associativity of the simulation to the design model

## 7 Works Cited

1. Clusterland Oberösterreich GmbH. Erhöhung der Standzeit und Reduktion der Taktzeiten. Kunststoff KC Cluster. [Online] [Cited: 05 12, 2012.] [http://www.kunststoff-cluster.at/files/Projektblatt\\_Antihaftbeschichtung.pdf](http://www.kunststoff-cluster.at/files/Projektblatt_Antihaftbeschichtung.pdf).
2. Kennedy, P.K. Practical and Scientific Aspects of Injection Molding Simulation. Eindhoven : Technische Universiteit Eindhoven, 2008. p. 155. ISBN 978-90-386-1275-1.
3. Plastifizieren in kürzester Zeit. WÜRTELE, DIPL.-ING. MARTIN, LANGE, DIPL.-ING. CHRISTOPH and HUNGERKAMP, DIPL.-ING. THOMAS. [www.kunststoffe.de/pe](http://www.kunststoffe.de/pe) : Kunststoffe, 2004, Vol. 06. PE102913.
4. ZYKLUSZEITEN GESENKT. Seidel, Franziska, Brunner, Dietmar and Wißuwa, Prof. Dr. Eckhard. s.l. : Plastverarbeiter, 2009, Vol. 10.
5. Multi-Physics Simulation Technology in NX. Ruel, Christian. Noordwijk, The Netherlands : ESTEC, 2011. 25th European Workshop on Thermal and ECLS Software.
6. Siemens PLM Synchronous Technology Part2/2. Siemens, 2008. <http://www.youtube.com/watch?v=f1Y8TEJd6iE&feature=related>.
7. Siemens Product Lifecycle Management Software Inc. NX 8 Hilfe CAE Advanced Simulation. [Online Help] s.l. : Siemens Product Lifecycle Management Software Inc., 2011.
8. Inc., Siemens Product Lifecycle Management Software. NX 8 Hilfe NX Nastran User Guide. [Online Help] s.l. : Siemens Product Lifecycle Management Software Inc., 2011.



Seminar:  
Computational Fluid Dynamics  
(CFD) in Systems Simulation

12 - 13 November 2013

Bamberg, Germany

ISBN: 978-1-874376-71-2  
EAN: 9781874376712



## Integration of complex 3d-simulations within systems simulations using response surfaces

Dr. Dirk Hartmann, Michael Mahler

Corporate Technology, Siemens AG, D-80200 Munich, Germany  
Siemens Industry Software GmbH & Co. KG, D-70565 Stuttgart, Germany

### Summary:

On the one hand, simulation of single components in technical systems, like 3D-FEM or 3D-CFD, can be considered today as classical engineering disciplines. The level of detail and complexity of such component simulations is increasing at an enormous speed. On the other hand, system simulation, modeling and predicting the behavior of complex technical systems, is a topic of increasing importance. However, the two modeling approaches addressing technical systems on different levels of detail are often not systematically connected.

In this contribution, we present a novel response surface based method allowing systematic connection of both approaches and reducing complex simulations. The approach is based on a sampling of detailed simulation snapshots within the parameter space of interest and interpolation of the functional / physical quantities of interest via interpolation methods. Using efficient sampling methods in combination with robust approximation schemes, e.g. moving least squares or kriging, allows an accurate reduction of complex models involving up to 20 independent parameters (and runtimes on the order of minutes / hours) to reduced models (runtimes in the order of seconds) within reasonable time scales. Reduced models can be accessed via standard interfaces, e.g. using the FMI-standard, and can thus be easily integrated in various system simulation tools.

As an example we consider a control for cooling of a power supply via air flows. Based on 3D CFD-simulations temperatures at critical electronic components are evaluated and the operation of the power supply is adjusted. Parameters are powers as well as air flow rates. Physical quantities of interest are temperatures at various sensor locations.

### Keywords:

System Simulation; Nonlinear 3D models; Model Order Reduction; Response Surfaces; Functional Mock Up Interface

1 Introduction

In the last years, a new trend is emerging: Simulation Driven Product Development. It aims at developing products along with a continuous simulation on a system as well as product level. The ultimate goal is a seamless virtual representation of the product on different levels - from a system description, in terms of a system-simulation, to single parts or assemblies [2][7], in terms of single highly detailed simulation models. On all levels an enormous advancement in simulation technology has been achieved in the last years. Mature commercial solutions are available on all different levels and are successfully used in many R&D departments. Especially on the system simulation level standardization [12] is advancing. Plug and simulates solutions combining models of different type and suppliers are no longer a vision but rather reality.

However, although we are coming closer and closer to the vision of a fully digital functional representation of products on all levels, models on different levels are quite often not linked systematically. On the one hand, automatic model generation aims at deriving detailed models on descriptions on different levels and could be used to automatically derive detailed component models. These techniques typically require a high degree of standardization in design rules, which is not always available. On the other hand, model order reduction [3] allows deriving fast executable models from complex slow detailed models. The corresponding fast models could be used e.g. within a system simulation. Model order reduction is well established for linear models, e.g.[1][3], but for non-linear models it is subject to active research in the simulation technology community [14][17].

In this contribution, we address a novel approach to complexity of non-linear systems based on response surface models. The approach allows us to systematically connect simulation models of different levels of detail, i.e. integration of a complex detailed simulation model within a system simulation. Due to the generality of response surface models, the approach is not restricted to any specific problem, physics or solvers. Combining the response surface model with a standardized interface for system simulation [12] the approach is completely solver independent. A schematic sketch of the approach is shown in Figure 1.

In the following sections, we first review the concept of response surface modeling (Section 2) and then address the aspect of interfacing different type of simulation models within system simulations (Section 3). We then present a demonstrator within NX [15] and illustrate the approach along a use case in Section (Section 4). Within this article, we restrict us to a quasi-static approach, but an extension to the dynamic case is possible and will be addressed in the discussion (Section 5).

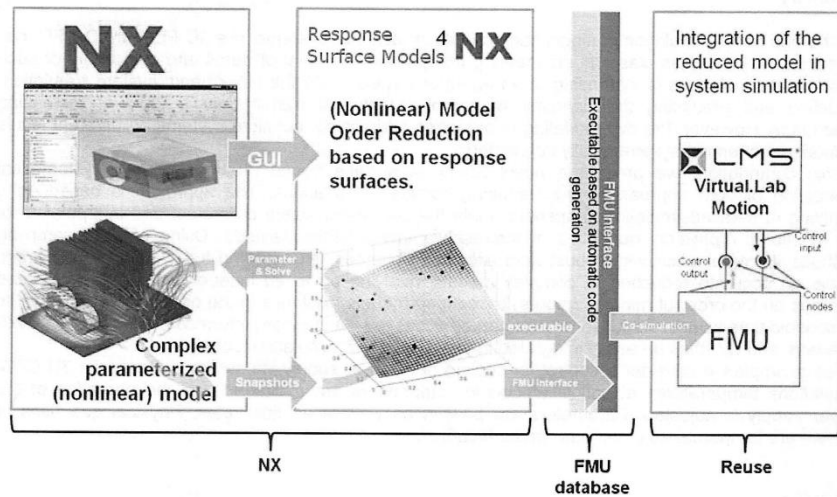


Fig. 1: Schematic sketch of the integration of complex 3d simulations into system simulations using response surface methods as a method for model order reduction. The reduced models support the Functional Mock-up Interface (FMI) co-simulation standard and thus can be integrated in any system simulation tool.



## 2 Response surface modeling

Modeling physical phenomena, e.g. a static analysis, one is typically not interested in all states of the detailed mathematical model but rather in the input-output relations, e.g. the relation of the deformation of a certain node (output) with respect to the applied force (input). Model order reduction aims at algorithmically reducing the complexity of mathematical models, e.g. the number of degrees of freedom in a FE model, while keeping the same predictive power. Model order reduction is subject to active research for quite some time, see e.g. [1][3] for an overview. Corresponding algorithms for model order reduction of linear systems have made their way into commercial products. For example, super-elements in FE solvers [15] are based on model order reduction techniques and even Matlab interfaces to reduced models are provided [3].

But when it comes to nonlinear systems model order reduction is subject to current research. Recently proposed approaches like reduced basis methods [14] or proper orthogonal decomposition [17] require an intrusive approach, i.e. need direct access to the underlying equations and their corresponding discretization, which is hardly accessible in commercial simulation products. Having in mind a model and solver independent approach, approximations of the original models, e.g. detailed finite element model, offers an attractive alternative to state-space based model order reduction. The probably most known method to approximate such relationships between input and output are response surface techniques dating back to the famous work of [4]. Corresponding methods to determine corresponding response surface methods are e.g. Kriging [9] or Moving Least square approximation [13], on which we rely here.

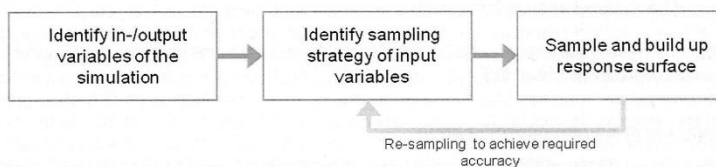


Fig. 2: Schematic sketch of the response surface modeling

In our context setting up a response surface model for a physical sub-system consist of three steps (c.f. Figure 2): First the right input, e.g. forces, and output variables, e.g. deformation of a point, of the subsystem have to be identified. These are typically significantly smaller than the number of internal states, e.g. values at FEM nodes. In the next step, a sampling strategy for the input variables is identified. The sampling itself is highly automated and does not require any user interaction. During the sampling the output variables are evaluated at the sampled input variables. Based on the samples a corresponding response surface model is reconstructed. The accuracy of the response surface can be evaluated a-posteriori by comparison of the reconstructed surface with the detailed simulations, i.e. additional simulations. Using this information, the last two steps could be repeated iteratively to obtain a response surface satisfying required accuracies.

The sampling strategy as well as the response surface method are outlined in more detail in the following two paragraphs.

### 2.1 Sampling

Sampling is the key to success in setting up a good response surface without using too many sampling points keeping in mind that evaluating a sample point could be quite expensive from a computational point of view. Choosing a good sampling approach is probably even more important than having the most advanced approximation method for the response surface. Each reduction of sampling points will reduce source investments and consumed time.

Let us assume for the moment, that we have  $n$  input variables  $y_i^l \leq y_i(t) \leq y_i^u$  with lower bounds and upper bounds. Sampling each interval with  $m$  equidistantly distributed values taking a naïve approach, one needs in total  $n^m$  samples. This could become rather large although for small numbers of variables and quite coarse sampling, e.g.  $5^5=3125$ . An alternative to this naïve approach would be a purely random sampling or Latin Hypercube sampling taking into account previously sampled points.

Within our approach, we rely on Latin Hypercube sampling [8][11], which samples in a random fashion but taking previously sampled points into account. To do so, the intervals of the  $n$  input variables, i.e.  $[y_i^l, y_i^u]$ , are distributed in  $m$  equally probable intervals.  $m$  sample points are then randomly distributed according to the Latin Hypercube requirement, that each sample is the only one in each axis-aligned

hyper-plane containing it (c.f. Figure). Thus by construction the scheme does not need more samples introducing more dimensions. Additionally one could impose the requirement that the entire space is sampled evenly leading the so-called orthogonal sampling scheme.

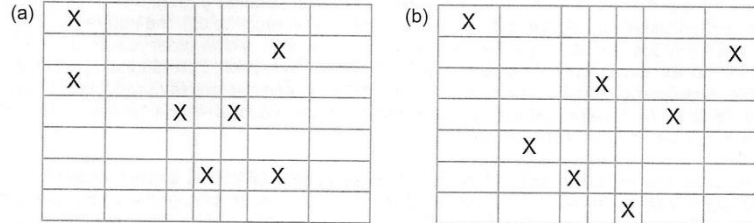


Fig. 3: Schematic sketch of a random (a) and a Latin Hypercube (b) sampling strategy for two independent input variables. Intervals of different length shall indicate different probabilities.

The sampling does not include a sensibility evaluation of the surface response, e.g. is independent of the detailed surface structure. To ensure that the relevant structure is covered, e.g. peaks with a small support in the input parameter space, either automatic post-sampling (c.f. Section below) or alternatively additional user-interaction might be necessary. For example, additional functionality allowing the user to influence the number of sampling points in a predefined range of the input variables could be realized straight forward [16].

Sampling routines like Latin hypercube are well established and are available in a wide range of statistical software packages, e.g. [16].

2.2 Moving Least Square approximation

A wide variety of response surface methods have been developed in the past. Moving least square approximations is probably the most established method when it comes to the reconstruction of continuous functions  $u(y)$  from a set of unorganized points  $y_i$  (in diverse applications ranging from particle methods [5] in continuum mechanics to computer graphics [13]).

Here we follow [5]. Moving least square interpolation looks at an arbitrary point  $y \in \mathbb{R}^n$  in the input space for a best interpolation of data  $u \in \mathbb{R}$  given at a (randomly) distributed set of points  $Y = \{y_i, i = 1..n\}$  in the form

$$u_{MLS}(y) = \sum_{i=1}^m p_i(y) a_i^*(y) \tag{4}$$

with  $p_i(y)$  given by a set of functions  $\mathcal{P}$  of size  $m$ , e.g. polynomials. In the following, we will also use roman indices to distinguish between the different functions belonging to  $\mathcal{P}$ . The coefficients  $a_i^*(y)$  are chosen such that the weighted least square error is minimised (as suggested by the name of the method), i.e.

$$a_i^*(y) = \arg \min_{a \in \mathbb{R}^m} \sum_{j=1}^n \left( \left( \sum_{i=1}^m p_i(y_j) a_i(y) \right) - u_j \right)^2 W(y - y_j, \eta). \tag{5}$$

Here,  $W(y, \eta)$  is an arbitrary positive weight function with compact support ensuring that only local information is used for interpolation. The parameter  $\eta$  controls the locality of the weight function. Since the coefficients  $a_i^*(y)$  depend directly on the position of evaluation  $y$  (the moving part of the name) an independent minimization problem has to be solved at each position  $y$  where the response surface shall be evaluated.

An explicit formula for the coefficients  $a_i^*(y)$  can be easily obtained by minimizing (5) leading to an interpolant of the form

$$u_{MLS}(y) = \sum_{j=1}^n \varphi_j(y) u_j \tag{6}$$

$$\varphi_j(y) = \sum_{k,l} p_k(y) A_{kl}^{-1}(y) B_{lj}(y)$$

with

$$A_{kl} = \sum_{j=1}^m p_k(y) p_l(y) W(y - y_j, \eta),$$



$$B_{ij} = p_i(y_j)W(y - y_j, \eta)$$

(for more details we refer to [5]).  $W$  would like to stress that the construction of the  $\varphi_j(y)$  is independent of the  $u_j$  and depends solely on the input variables. The interpolation is therefore independent of the output dimension. For an evaluation of the interpolant  $u_{MLS}$  at a position  $y$  for each shape function one needs to invert the square matrix  $A$  of size  $m \times m$ , where  $m$  is the size of the function space  $\mathcal{P}$ .

Once a functional surface has been set-up errors can be evaluated by comparing values from the response surface model and evaluation of the detailed simulation model. This allows the identification of regions where additional samples would be necessary to achieve required accuracies. In these regions additional samples can be evaluated. Since each evaluation of the response surface implies the solution of (5) on the basis of available sample points no further effort or user interaction is required.

Many software packages are available providing moving least square interpolation, e.g. [6].

### 3 Interfacing with other simulations

The vision of a standardized interface for computer simulations of complex cyber-physical systems has been pushed forward by the European consortium MODELISAR established in 2008 [12]. It pushes the idea of assembling a virtual product from separate models of components - just like real cyber-physical systems are assembled - forward. Thus, it aims at providing the basis for model based development of systems. The standard has been first published in 2010 and continuous development of the standard is today driven by more than 15 companies and research institutes and is supported by a large variety of tools. The FMI standard supports two different aspects: On the one hand it, supports the exchange of models described by differential, algebraic and discrete equation in terms of solver-independent C-code that can be used with a solver of another party, referred to as FMI for Model Exchange. On the other hand, it provides an interface for coupling two or more simulation tools communicating at discrete communication points. The organization of the co-simulation slaves, which solve the corresponding models independently of each other, are coordinated by a so-called simulation master, which could simultaneously also provide a simulation slave. This is referred to FMI for co-simulation.

Within our approach we will rely on the FMI for co-simulation standard [12] providing at the same time the model and the corresponding solver. Principally, the standard supports two ways for providing the corresponding code: either a single component including the model and its solver or two separate components. The interface is based on a slender set of simple C-functions and an additional XML-file including the specific properties of the slave component as well as a set of capability flags. The master algorithm orchestrating the simulation is not part of the standard.

A simulation model taking part in a system simulation could be characterized by a set of internal state variables. These are connected at discrete communication steps  $tc_i$  to other simulation-slaves using inputs and output ports of the FMI. The physical connections between subsystems are represented by mathematical coupling conditions between the inputs and the outputs of all subsystems [10]. During the steps  $tc_i \rightarrow tc_{i+1}$  the subsystems are solved independently using their associated solvers. The standard covers details of the data exchange between the sub-systems as well as algorithmic issues to synchronize the simulation of all subsystems and to proceed in communication steps (macro steps)  $tc_i \rightarrow tc_{i+1}$  from initial time  $tc_0 := t_{start}$  to end time  $tc_n := t_{end}$  [12]. For more details we refer to the original standard [12]. The combination of the response surface and FMI standard could achieve the interfacing of the response surface to other simulations.

System models are usually set up manually but automation is achievable. Combining response surface techniques with standardized interfaces for system simulation allows deriving in a highly automated fashion system models independent of tools and solvers. Sampling the input parameter space response surface models can be setup coupling the input and output relationship of the physical subsystem under consideration.

Within this contribution we restrict us to quasi-static sub-systems, i.e. assuming that the underlying dynamics of the sub-system are significantly faster than those of the total system. Thus dynamics can be neglected and the input-output relation is static, i.e. independent of communication step sizes  $tc_{i+1} - tc_i$ . A generalization to sub-systems with truly dynamic behavior is possible and is

addressed in the discussion. A prototypic implementation has been realized in the NX-tool suite using the NX-open functionality (c.f. also Figure 1) and is demonstrated below.

**4 Prototype, demonstrator and use case example**

The GUI of the prototype is fully integrated in the NX tool suite via the NXOpen functionality. After loading a simulation solution which has been set up before as a detailed model, input parameters for the system simulation are identified. Similarly, the corresponding output values have to be identified and set up such that they are available from a scripting basis. In the next step the user identifies the input parameter ranges which are relevant with respect to the system simulation. Furthermore, the user can choose additional options like sampling rates or response surface methods. The sampling is then performed automatically by the prototype. This can take a significant amount of time dependent on the complexity of the detailed simulation as well as the required accuracy. But the process itself is highly automatic and does not require any user interaction. At the end a functional mock-up unit (FMU), i.e. an independent simulation unit according to the FMI standard, is automatically generated according to the standard packed in a zip-file.

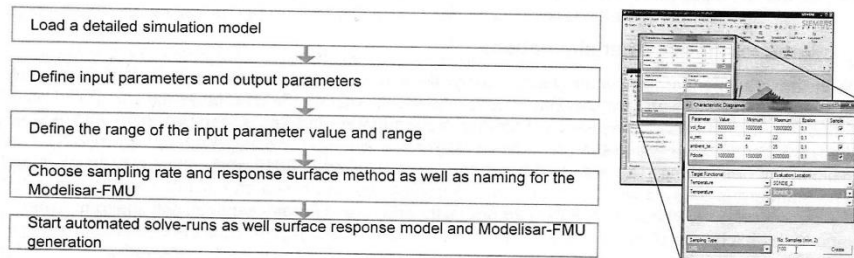


Fig. 4: Workflow of the NX prototype

The functional mock-up unit includes the input and output values of the sampled model as well as an executable response surface interpolator acting as the solver of the functional mock-up unit. It can be integrated as a standard functional mock-up unit in more than 30 tools supporting the FMI standard, e.g. Matlab or AMESim. In contrast to the evaluation of the detailed model (which can be in the order of minutes or even hours) the evaluation of the response surface model is in the order of seconds or even milliseconds.

As a use case to demonstrate the concept we consider the temperature control of an electronic power unit, e.g. to protect sensitive electronic devices or parts (c.f. Figure 5a).

The electronic equipment in the power unit heats up the case. The power unit can be cooled with a ventilator providing an air stream with a temperature at the inflow defined by the ambient temperature. The ambient temperature varies, e.g. due to daily changes, and cannot be controlled. However, the electric power of electronic components as well as the ventilator speed can be controlled. The temperature inside the case is measured with two probes and can be used as the basis for the control. Goal of the control is to keep the temperature at the probe close to 67° C. Using detailed CFD simulations, airflows and temperatures can be predicted (c.f. Figure 5b) and the control can be tested on the bases of corresponding models.

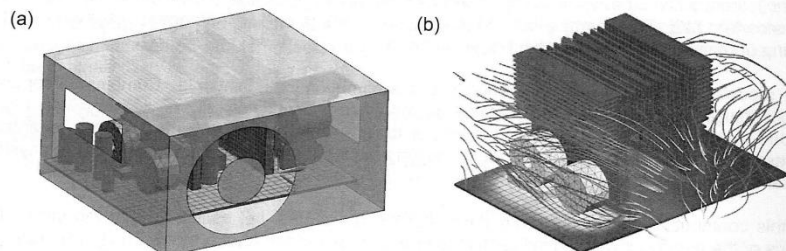


Fig. 5: Use case: Design of a cooling control of an electronic power unit (a) by means of a CFD simulation (b). Goal is to adjust the fan or power such that the temperature at a probe is smaller than a given threshold.



Using a detailed CFD model requires enormous computation times and therefore the approach of testing the control on the basis of simulation would be nearly impossible apart from missing FMI interfaces in most 3D simulators – of course here we consider a simple test example and setting up such a simple control actually would require no simulations at all. Adopting the concept introduced above the design of the control along efficient system models is well feasible. A corresponding system model with a FMI interface is therefore derived using the developed prototype.

The functional mock up unit has been imported in the in-house system simulation platform COSMOS [18] and a very simple control has been designed (c.f. Figure 6). The control considers only one temperature sensor as an input and controls only the power of the electronic equipment. In Figure 7 one simulation of the response to a rising ambient temperature is shown.

This toy example can be generalized to more complex physics and application, ranging from first validation to virtual commissioning or even operation based on system simulation. Especially input parameters could be of any type, e.g. also parameters of the detailed model's underlying geometry (like thickness or radii). Since it is completely tool independent an integration into other system simulators is straight forward. An extension of the prototype to different simulation tools rather than the NX tool suite is as well possible, as long as a sufficient environment for scripting the simulation tools is provided in order to integrate the sampling, which is the case for most available tools.

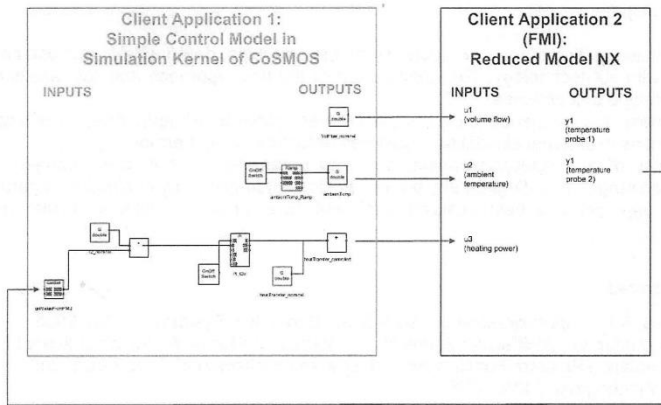


Fig. 6: Design of a control by means of system simulation. Communication in Co-simulation is handled by Cosmos middleware [18] used as the system simulation platform.

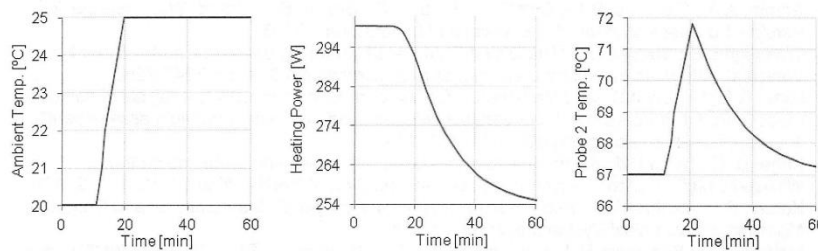


Fig. 7: A rise in the ambient temperature leads to a decrease in power ensuring that the temperature at the probe 2 does not exceed the required threshold over a longer time.



## 5 Discussion

Within this contribution we have outlined a new approach to integrating detailed simulation models in a systems simulation. It combines the following steps:

- GUI guided pre-processing for the generation efficient system models
- Automated cycle how will including the
  - o Sampling of input variables (Latin hypercube sampling)
  - o Solve run of the simulation model at sample parameters
  - o Construction of the response surface (moving least square approximation)
  - o Post sampling where the accuracy of the response surface does not meet the requirements
- Automated generation of an standardized FMU allowing easy interfacing of the response surfaces

The generated FMU will be a fast solving simulation model which requires significantly less recourses than the original simulation model. Additional the FMU could interface with other simulation models and provides a fast and easy integration to system-simulation models. What could be used to support a system driven development process. The sharing of the standardized FMU will keep knowledge and technology of the simulated object secret. Such sharing could be with other departments or also companies integrating the FMU into their virtual prototypes. This will reduce the number of development cycles, development resources and will lead to faster development process as well as an improved development quality.

The validation and testing of the new approach by the prototype, demonstrator and use case example has realized with NX-technology. The combination of the new approach and NX results in a flexible and open prototype that provides

- openness and independence in the used solvers (Nastran, Abaqus, Ansys, LSDyna, TTM, ...)
- openness in different simulation disciplines (structure, flow, thermal, ...)
- usability of any input-parameters, e.g. also parameters that causes changes (in the pre-defined range) in CAD-geometry as well as simulation-geometry or simulation-parameters

Although the approach has been realized within NX an extension to other simulation tools can be realized easily.

## 6 References

- [1] Antoulas, A.C.: "Approximation of Large-Scale Dynamical Systems", SIAM, 2005
- [2] Bals, J.; Hofer, G.; Pfeiffer, A.; Schallert, C.: "Virtual Iron Bird – A Multidisciplinary Modelling And Simulation Platform For New Aircraft System Architectures", DGLR Luft- und Raumfahrtkongress 2005, 2005.
- [3] Bechthold, T.: "Introduction into Model Order Reduction", CADFEM Journal, 2012, 53-55
- [4] Box, G. E. P. and Wilson, K.B. (1951) On the Experimental Attainment of Optimum Conditions (with discussion). Journal of the Royal Statistical Society Series B13(1):1–45
- [5] Dilts, G.A.: "Moving-least-squares-particle hydrodynamics - I. consistency and stability", Int. J. Numer. Meth. Eng. 44, 1999, 1115-1155
- [6] Giunta, A.A.; Richards, M.D.; Cyr, E.C.; Swiler, L.P.; Brown, S.L. Eldred, W.S.: "Surfpack Version 1.0 User's Manual", Sandia National Laboratories, 2006
- [7] Glaessgen, E.; Stargel, D. : "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles", American Institute of Aeronautics and Astronautics 8; 2012, 7247-7260
- [8] Iman, R.L.; Helton, J.C.; and Campbell, J.E.: "An approach to sensitivity analysis of computer models, Part 1. Introduction, input variable selection and preliminary variable assessment". Journal of Quality Technology 13, 1981, 174–183.
- [9] Krige, D. G.: "A statistical approach to some basic mine valuation problems on the Witwatersrand". J. of the Chem., Metal. and Mining Soc. of South Africa 52, 1951, 119–139
- [10] Kübler, R.; Schiehlen W.: "Two methods of simulator coupling", Mathematical and Computer Modeling of Dynamical Systems 6, 2000, 93-113.
- [11] McKay, M.D.; Beckman, R.J.; Conover, W.J.: "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". Technometrics (American Statistical Association) 21, 1979, 239–245
- [12] Modelica Association: "FMI for Co-Simulation", Version 1.0, 2010
- [13] Nealen, A.: "An As-Short-As-Possible Introduction to the Least Squares, Weighted Least Squares and Moving Least Squares Methods for Scattered Data Approximation and Interpolation", Tech. Rep. TU Darmstadt, 2004

- [14] Patera, A.T., Rozza, G.: "Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations", MIT Pappalardo Graduate Monographs in Mechanical Engineering, 2007
- [15] Siemens PLM Software: "NX 9.0 for simulation", [http://www.plm.automation.siemens.com/en\\_us/products/nx/for-simulation](http://www.plm.automation.siemens.com/en_us/products/nx/for-simulation), 2013.
- [16] Swiler, L.P.; Wyss, G. D.: "A User's Guide to Sandia's Latin Hypercube Sampling Software: LHS UNIX Library/Standalone Version", Sandia National Laboratories, 2004.
- [17] Volkwein, S.: "Model Reduction using Proper Orthogonal Decomposition", Lecture notes, University of Constance, 2011
- [18] Wöhlhaf, K.; Rosen, R.: "A component-oriented simulation approach for industrial plant models: The PlanSim simulation tool", ESM 2000, Ghent, 2000, 291-295.

## 9.5 APPENDIX E PATENT CONFIRMATION

*The general concept of the invention will be organisation regulars of simulation data including regulars to organise data and to relate data. These organisation regulars will be the core technology. The implementation of the organization regulars into TEAMCENTER could enhance TEAMCENTER to support and manage multi- and inter-disciplinary simulation. The organisation regulars achieve an enhanced representation of relations between simulation, product and function data. In addition the organisation regulars achieve an enhanced simulation result data handling. So, the invention makes relationships such as sourcing, targeting, relating and sub ordering of data in a multi- and inter-disciplinary simulation uniquely traceable. The patent should protect these organization regulars.*

*The invention addresses issues in the simulation of complex products and processes. The simulation of such products and processes will be limited because of the complexity of the products, or processes will be high volume, and require interaction between multiple simulation models. The generation and interaction of such multi- and inter-disciplinary simulations will be limited because there will be a gap between the data management and support of such data by a company, department, etc. overlapping way.*

*The invention of the organisation regulars and the implementation of the organisation regulars into TEAMCENTER provide a huge enhancement in the support and management of multi- and inter-disciplinary simulation. So, the generation of such simulation will get supported and managed in a PLM environment.*

**SIEMENS****Industry**Herrn  
Michael Mahler

Unser Zeichen	2013E02294 DE / 2013P06210US
	EVE / SEY
Datum	05.Feb.2014

**Erfindungsmeldung vom 05.02.2013, Eingang: 05.02.2013**  
**Erfinder: Herr Michael MAHLER, geb. 03. Juli 1977**

Titel: "Verfahren bzw. Organisationsregeln zum Verwaltung und Managen von multidisziplinären Simulationsdaten sowie Verfahren bzw. Organisationsregeln zum Unterstützen von interdisziplinären Systemsimulationen"

**BESTÄTIGUNG**

Hiermit bestätigen wir, dass Herr Michael Mahler die o.g. Erfindung an die Siemens Industry Software GmbH & Co. KG gemeldet hat.

Die Erfindung wurde am 21.3.2013 beim Patentamt der U.S.A. (USPTO – United States Patent and Trademark Office) als Patentanmeldung eingereicht; der Patentanmeldung wurde dabei das amtliche Aktenzeichen 61/803,933 zugewiesen. Das USPTO erteilte dazu am 1. Mai 2013 eine Foreign Filing License.

[REDACTED] Erfindung ein wertvoller technischer Beitrag gesehen wird. [REDACTED]

Es bestehen diesseits keine Einwände, dass Herr Mahler den o.g. Gegenstand als Bestandteil einer wissenschaftlichen Publikation, insbesondere als Bestandteil seiner Dissertation zur Erlangung der Doktorwürde, veröffentlicht.

Die Anmeldebestätigung des USPTO und eine Kopie der Erfindungsmeldung liegen dieser Bestätigung bei.

SIEMENS INDUSTRY SOFTWARE GMBH &amp; CO. KG

*[Signature]*  
 [REDACTED]  
 Prokurist

*[Signature]*  
 [REDACTED]  
 Head of Human Resources

Anlage  
 Filing Receipt (USPTO)  
 Erfindungsmeldung vom 5.2.2013 (Kopie)  
 Siemens Industry Software GmbH & Co. KG

Franz-Geuer-Straße 10  
 50823 Köln  
 Deutschland

Tel.: +49 (221) 20802-0  
 Fax: +49 (221) 248928  
 www.siemens.com/plm

Kommanditgesellschaft: Sitz der Gesellschaft: Köln; Registergericht: Amtsgericht Köln, HRA 28227;  
 Geschäftsführung und persönlich haftender Gesellschafter: Siemens Industry Software Management GmbH;  
 Geschäftsführer: Urban August; Daniel Trebes  
 Sitz der Gesellschaft: Köln; Registergericht: Amtsgericht Köln, HRB 70858





UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
 United States Patent and Trademark Office  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
61/803,933	03/21/2013		260	2013P06210US		

CONFIRMATION NO. 1816

FILING RECEIPT



45113  
 Siemens Corporation  
 Intellectual Property Department  
 170 Wood Avenue South  
 Iselin, NJ 08830

*Eve*

MAY 10 2013

Date Mailed: 05/06/2013

Receipt is acknowledged of this provisional patent application. It will not be examined for patentability and will become abandoned not later than twelve months after its filing date. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Michael Mahler, Ketershausen, GERMANY; ✓

Applicant(s)

Michael Mahler, Ketershausen, GERMANY;

Power of Attorney:

Andrea Evensen-56531

If Required, Foreign Filing License Granted: 05/01/2013

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 61/803,933**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No

Title

MULTI-DISCIPLINARY SYSTEM-SIMULATION IN A PLM ENVIRONMENT ✓

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

**PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES**

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same

effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

#### **LICENSE FOR FOREIGN FILING UNDER**

#### **Title 35, United States Code, Section 184**

#### **Title 37, Code of Federal Regulations, 5.11 & 5.15**

#### **GRANTED**

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with

page 2 of 3

respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

**NOT GRANTED**

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

---

**SelectUSA**

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.