Vorstellung einer generischen und unabhängigen XML-basierten physikalischen Beschreibungssprache "SMILE"

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

Berlin, 16.12.2020

Hochschule für Technik und Wirtschaft Berlin
Fachbereich 2, Fahrzeugtechnik
Darius.Friedemann@HTW-Berlin.de / +49 30 5019-3676
Joerg.Rademann@HTW-Berlin.de / +49 30 5019-3854

Einleitung

In der Entwicklung von Kraftfahrzeugen und deren Komponenten nimmt der Anteil der anhand von computernumerischen Simulationen durchgeführten Untersuchungen seit Jahren stetig zu. Dies liegt sowohl an den immer komplexer werdenden Anforderungen an die Komponenten, als auch an den sich immer weiter verkürzenden Entwicklungszyklen von Kraftfahrzeugen. Des Weiteren bieten Simulationen in diversen Entwicklungsbereichen große Kostenvorteile im Vergleich zum Aufbau und Testen von Prototypen.

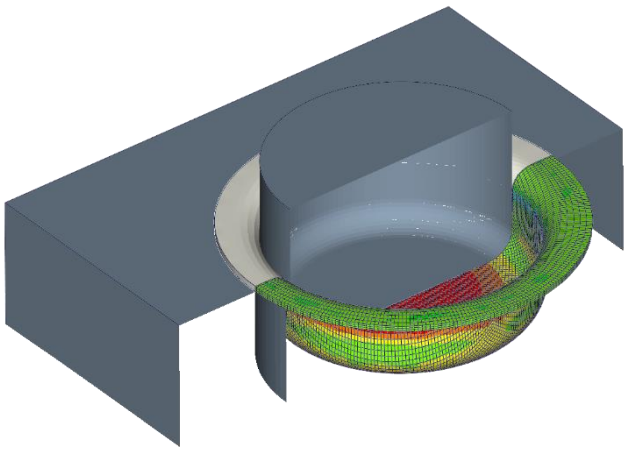

Abbildung 1: Simulation eines Tiefziehprozesses

Aufgrund der Vielzahl von unterschiedlichen Entwicklungsbereichen von Fahrzeugkomponenten ist es heutzutage üblich, eine Komponente mit verschiedenen Softwaretools zu modellieren und zu testen. Die Verwendung unterschiedlicher Tools begründet sich teilweise durch verschiedene Ergebnisinteressen an einen Belastungsfall (z.B. NVH versus Spannungsnachweis) oder durch unterschiedliche Stärken und Schwächen der

einzelnen Softwaretools. Rückblickend betrachtet wurden in den letzten Jahrzehnten viele verschiedene Simulationsprogramme gleichzeitig entwickelt, welche jeweils für einen bestimmten Anwendungsfall optimiert wurden. Es ist für die nahe und mittelfristige Zukunft absehbar, dass in dem gesamten Entwicklungsprozess einer Komponente auch weiterhin unterschiedlichste Softwarelösungen verwendet werden. Die Tatsache, dass im Laufe der Zeit, viele Programme zu multidisziplinären Softwarelösungen weiterentwickelt wurden, die für die unterschiedlichen Anwendungsfälle nutzbar sind, wird daran vermutlich nichts Grundlegendes ändern.

Eine Eigenschaft haben alle Programme gemeinsam. Vom Anwender wird ein umfangreiches Verständnis von numerischer Simulation im Allgemeinen, der jeweiligen Simulationssoftware im Speziellen und fachspezifisches Wissen in Bezug auf das Anwendungsgebiet benötigt. Beim Wechsel auf ein anderes Programm innerhalb der gleichen Entwicklungsdisziplin, muss der Anwender zunächst ein umfangreiches, softwarespezifisches Spezialwissen aufbauen, weshalb eine Modellerstellung zunächst mit steigendem Aufwand verbunden sein wird. Das Fachwissen in Bezug auf ein Anwendungsgebiet und die numerische Simulation im Allgemeinen reichen für die Modellerstellung nicht mehr aus. Der Berechnungsingenieur(m/w/d) sollte für eine erfolgreiche Tätigkeit also idealerweise Experte(m/w/d) auf drei Gebieten gleichzeitig sein.

Das notwendige Wissen der einzelnen Teilgebiete Softwareverständnis, numerische Simulation und fachspezifisches Entwicklungs-Knowhow zu trennen, um den Fokus auf die physikalische Problemstellung zu lenken, bietet nicht nur die Möglichkeit Entwicklungszeit zu sparen (-> „Demokratisierung der Simulation“). Auch können Spezialisten für die jeweiligen Fachgebiete eingesetzt werden, die in Ihrem jeweiligen Bereich besonders gut qualifiziert sind. Somit kann die Entwicklungsqualität gesteigert werden. Hierzu ist eine standardisierte softwareunabhängige Sprache erforderlich, welche ausschließlich die physikalisch motivierte Problemstellung beschreibt. Dies wird im Folgenden gezeigt.

Ein Ansatz um das beschriebene Problem zu lösen, zeigt die Metasprache „Unified **S**imulation **M**odell**I**ng **L**anguag**E**“–**SMILE**. Diese stellt eine syntaktische Basis zur Verfügung, um eine Komponente nur hinsichtlich Ihrer physikalischen Eigenschaften zu beschreiben. Das spezifische Wissen bezüglich einer speziellen Simulationssoftware und der numerischen Simulation im Allgemeinen werden hierbei noch nicht benötigt. Sobald ein Simulationsmodell mit SMILE aufgebaut wurde, ist es möglich, dieses automatisiert in die Modellierungssprache eines beliebigen Solvers zu übersetzen und die zugehörige Simulation zu starten. Die Informationen über Modellierungsvorgaben und/oder Einstellungen, die der Solver selbst benötigt, können in SMILE-Modellierungsrichtlinien definiert werden. Diese

Modellierungsrichtlinien werden von Simulationsexperten der jeweiligen Softwaresysteme gepflegt. Die Übersetzung von Materialkarten für unterschiedliche Solver und Anwendungsgebiete ist prinzipbedingt nicht möglich und setzt zumeist Einzelfallvalidationen voraus. Daher wird bei der Modellierung in SMILE bewusst eine vorhandene Materialdatenbank für die verwendeten Solver vorausgesetzt.

Das automatisierte Überführen eines SMILE Modells in das Format eines Solvers, übernehmen Übersetzer, die scriptbasiert laufen oder in Preprozessoren eingebunden werden können. Diese werden für jede Software zur Verfügung gestellt. Für den Solver ABAQUS und LS-DYNA sind im Rahmen eines Forschungsprojektes bereits Übersetzer erstellt worden, um Machbarkeitsnachweise durchzuführen.

In Abbildung 2 sind die unterschiedlichen Entwicklungsprozesse mit und ohne SMILE gegenüber gestellt. Ausgehend von einem digitalen Prototyp, wird im aktuellen Prozess für

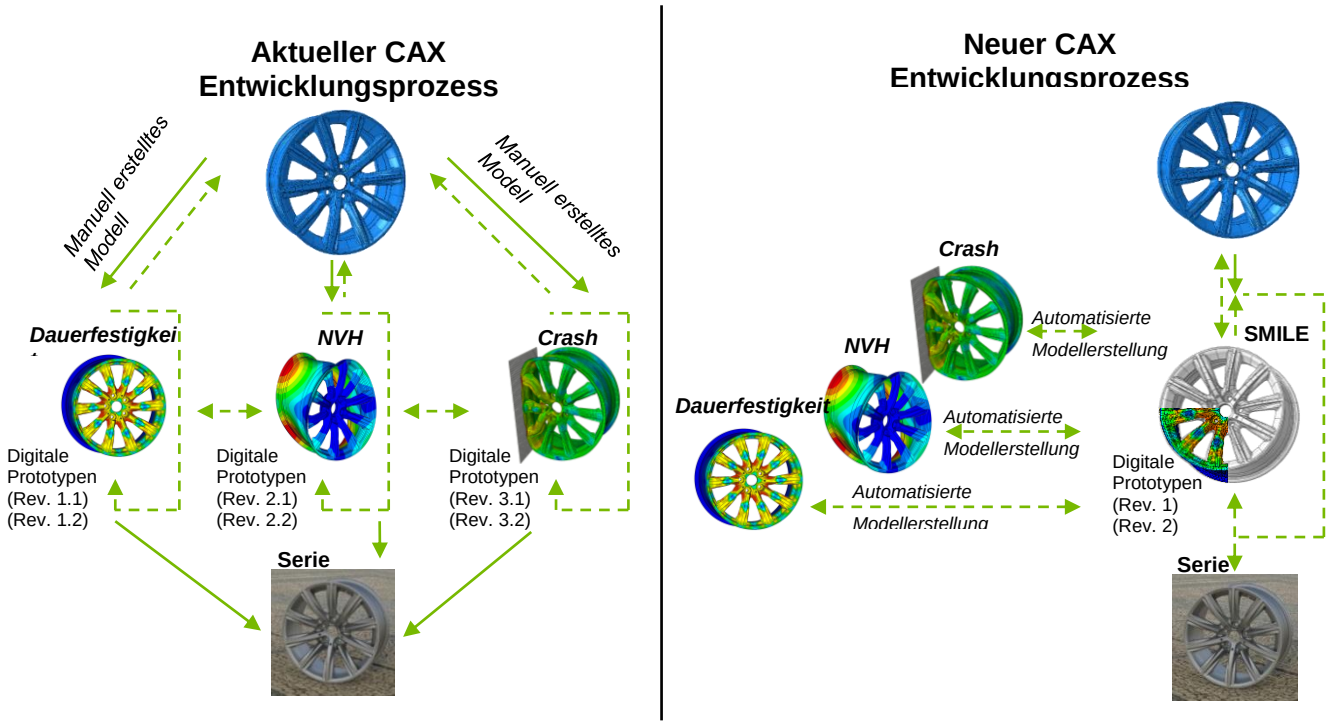

Abbildung 2: Vergleich der Entwicklungsprozesse ohne und mit SMILE

jede Entwicklungsdisziplin ein eigenständiges Simulationsmodell erstellt. In den einzelnen Disziplinen wird das Modell hinsichtlich der verschiedenen Anforderungen optimiert. Die unterschiedlichen Optimierungsergebnisse führen zu einer Kompromisslösung und somit zu einer Revision des digitalen Prototypens. Dieser iterative Entwicklungsprozess wird wiederholt, bis alle Anforderungen an den Prototypen erfüllt sind.

Der Entwicklungsprozess mit SMILE nutzt die Informationen des digitalen Prototyps. Zusätzlich werden in dem SMILE Modell alle Eigenschaften der unterschiedlichen Entwicklungsdisziplin definiert. Von diesem standardisierten Modell werden die verschiedenen Simulationsmodelle automatisiert erstellt und optimiert. Neue Designvorschläge werden als Revision im SMILE Modell hinzugefügt und eine erneute automatisierte Simulationsmodellerstellung kann erfolgen. Dieser Prozess wird wiederholt, bis alle Anforderungen an den Prototyp erfüllt sind. Bezogen auf den beschriebenen Entwicklungsablauf, zeigt Abbildung 3 grafisch das Potential, den Arbeitsaufwand durch die Verwendung von SMILE zu reduzieren. Ein Mehraufwand bei geringer Anzahl von Entwicklungsdisziplinen (zumeist nur bei n≤2) erklärt sich aus dem leicht erhöhten

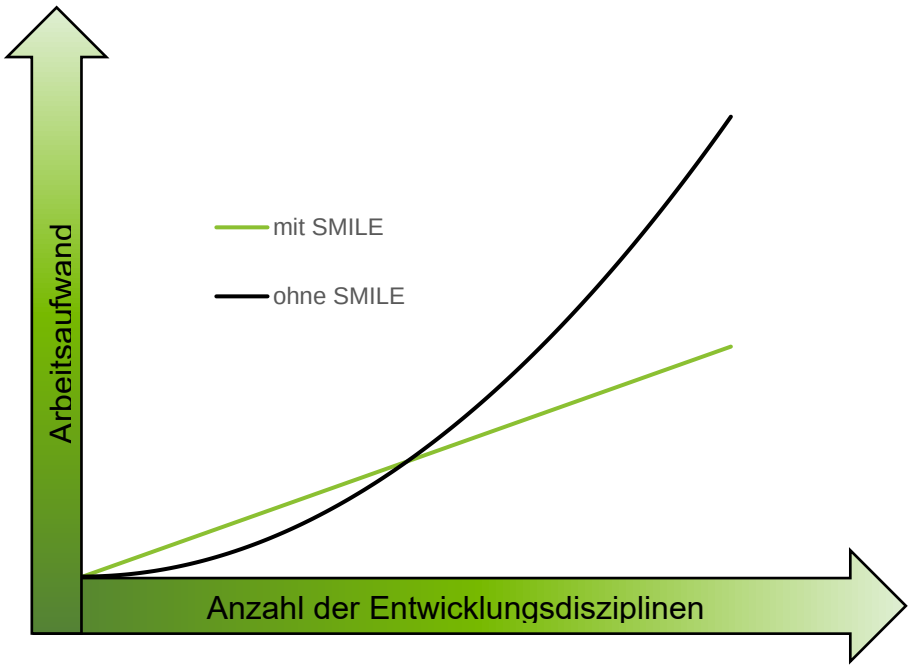

Abbildung 3: Gegenüberstellung der Entwicklungsprozesse

Modellierungsaufwand für die Erstellung von maschinenlesbaren Modellierungsrichtlinien. Bei steigender Anzahl von Entwicklungsdisziplinen wird der Mehraufwand bei der Modellierung durch die automatisierte Modellerstellung schnell kompensiert und der Arbeitsaufwand somit reduziert.

Aufbau und Ablauf

Die Syntax von SMILE basiert auf XML, da es sich hierbei um eine vollständig standardisierte und betriebssystemunabhängig lesbare Sprache handelt. Zusätzlich kann die Syntax einer

XML Datei mit den meisten Texteditoren geprüft werden. Weitere Vorteile von XML sind die ständige Weiterentwicklung der Sprache, die hohe Akzeptanz und die Tatsache, dass die Sprache nicht nur maschinenlesbar, sondern auch menschenlesbar ist.

SMILE bietet einen sprachähnlichen Ansatz um Simulationsmodelle zu beschreiben. Die Umfänge, die heutzutage mit einer numerischen Simulation abgebildet werden sind vielseitig, weshalb bei der Entwicklung von SMILE von vornherein viel Wert auf einen offenen und leicht erweiterbaren Standard gelegt wird. Mit SMILE können drei verschiedene Dateitypen erstellt werden: Modelldateien, Konfigurationsdateien und Modellierungsrichtlinien. Die SMILE-Modelldatei besteht aus einer Baugruppe und ggf. Unterbaugruppen die aus einer oder mehrerer Komponenten bestehen können. Alle physikalischen Eigenschaften der Baugruppe können unabhängig von der genauen Form einer Geometrie definiert werden, da in einer SMILE-Modelldatei mehrere FE-Netze oder auch CAD Geometrien referenzierbar sind. Wenn in einem Lastfall verschiedene Modelldateien benötigt werden, können diese in einer SMILE-

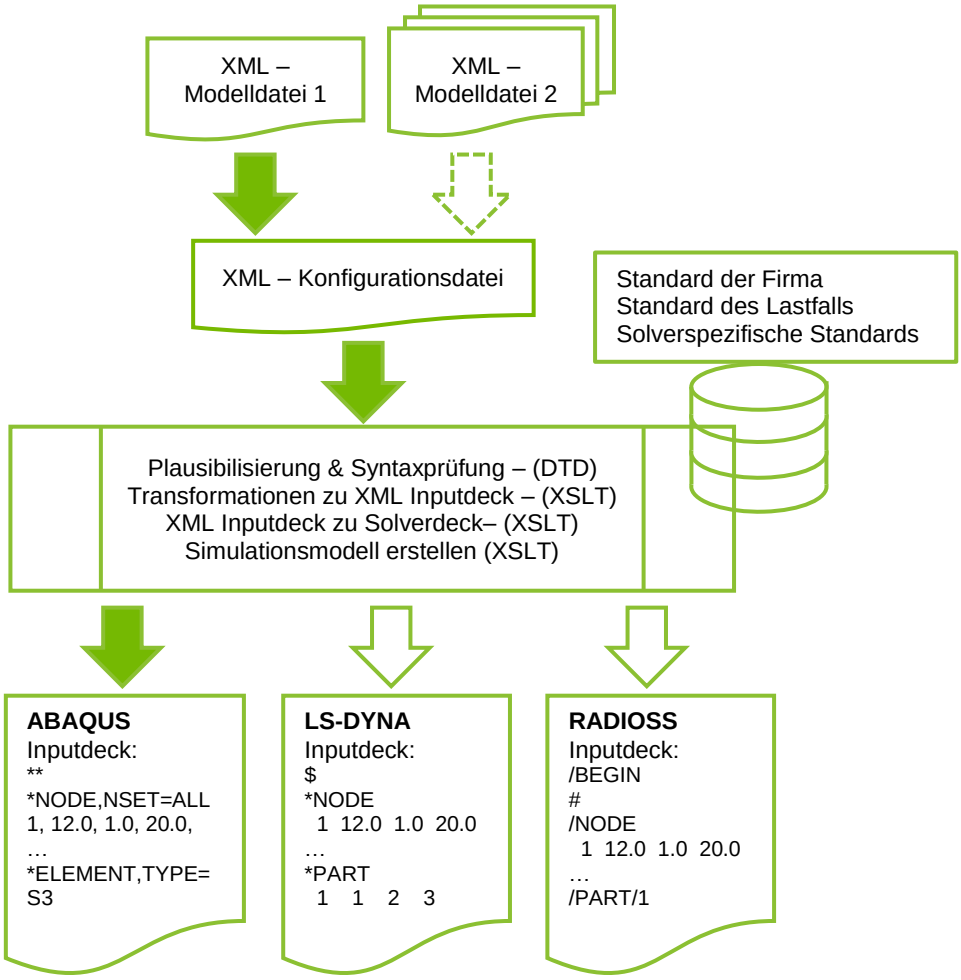

Abbildung 4: Gesamtübersetzungsprozess eines SMILE Datensatzes zu einem solverspezifischen Inputdeck

Konfigurationsdatei zusammengeführt werden. In dieser wird zusätzlich der Lastfall selbst, z.B. durch Anfangs- und Randbedingungen definiert. Damit sich der Entwickler(m/w/d) nicht mit solverspezifischen Einstellungen oder Idealisierungen befassen muss, werden diese in den SMILE-Modellierungsrichtlinien als maschinenlesbare Vorlage vordefiniert. Im Entwicklungsprozess können in Abhängigkeit vom Anwendungsfall unterschiedliche Idealisierungen benötigt werden, weshalb es möglich ist, verschiedene SMILE-Modellierungsrichtlinien für unterschiedliche Lastfälle oder Anwendungsgebiete zu nutzen. Diese können in unterschiedlichen Hierarchiestufen definiert werden, um die Vorgaben von unterschiedlichen Parteien zu berücksichtigen. Beginnend bei den Vorgaben vom Solverhersteller, folgen Vorgaben der Firma und abschließend Vorgaben der Fachabteilung. Die SMILE-Modellierungsrichtlinien stellen eine Grundlage zur Verfügung, um die unterschiedlichen Modellierungsansätze der unterschiedlichen Anwendungsgebiete bei der automatisierten Übersetzung zu berücksichtigen. Da keine solverspezifischen Bezeichnungen aus der FEM in einer SMILE–Modelldatei genutzt werden, jedoch trotzdem Idealisierungen vorgenommen werden müssen, kann auf die vom Softwarehersteller empfohlenen Richtlinien voreingestellt zurückgegriffen werden. In diesen sind beispielsweise Elementtyp unter Berücksichtigung des Anwendungsfalls, Elementeigenschaften und Kontakteigenschaften vordefiniert. Jedem Anwender ist es jedoch freigestellt, für den eigenen Lastfall abweichende Standardeinstellungen zu verwenden. Mit SMILE können auch eigene Modellierungsrichtlinien erstellt werden. In Abbildung 4 ist dargestellt, wie mit den unterschiedlichen SMILE Dateien ein Prozess zur automatisierten Übersetzung in Simulationsmodelle aussehen kann. Das Prozessablaufdiagramm zeigt die unterschiedlichen Dateitypen (Modelldatei, Konfigurationsdatei und Modellierungsrichtline) und wie diese im Gesamtprozess miteinander interagieren.