

FunctionalDMU: Co-Simulation mechatronischer Systeme in einem DMU-Umfeld

Autoren:

André Stork, Mathias Wagner, Fraunhofer IGD

Peter Schneider, Olaf Enge-Rosenblatt, André Schneider, Fraunhofer IIS/EAS

Andreas Hinnerichs, Carsten Neumann, Fraunhofer FOKUS

Thomas Bruder, Christiane Schäfer, Fraunhofer LBF

Einleitung

DMU-Systeme wie z.B. die TC Vis-Produkte von Siemens UGS oder der DMU-Navigator von Dassault Systèmes sind in der Industrie weit verbreitet. Sie sind gekennzeichnet durch die Verarbeitung multipler CAD-Formate (Multi-CAD) und Funktionalitäten zur geometrischen Analyse von Modellen, insbesondere der Kollisionserkennung. Was in DMU jedoch weitgehend fehlt, ist die Möglichkeit Verhalten und Funktion zu erleben.

In Analogie zur Modellierung von Geometrie in unterschiedlichen Softwaresystemen, werden Verhaltens- und Funktionsmodelle im virtuellen Produktentwicklungsprozess mechatronischer Komponenten ebenfalls in einer Vielzahl von Systemen erstellt und ausgeführt. Bekannte Vertreter solcher Systeme sind Matlab/Simulink, Modelica-basierte Systeme (Dymola, SimulationX), VHDL-AMS-fähige Systeme (z.B. Saber), sowie Adams, SimPack, Rhapsody, etc.

Diese Systeme haben ihre Einsatzschwerpunkte und sind als Einzelwerkzeuge eingesetzt nicht in der Lage, alle Fragestellungen der Gesamtsimulation mechatronischer Systeme zufriedenstellend zu beantworten. Gleichmaßen steigt der Druck nach frühzeitiger virtueller Gesamtintegration in der Industrie an, was die Systemanbieter zu mehr Offenheit drängt und die Nachfrage nach innovativen Lösungsansätzen erhöht.

Anstatt nun die n+1. direkte Kopplung von Simulationswerkzeugen umzusetzen, hat das Fraunhofer-Projekt FunctionalDMU (FDMU) das Ziel verfolgt, eine offene und herstellerunabhängige Plattform zur Einbindung unterschiedlichster Simulatoren aus den Domänen Software, Elektrik/Elektronik und Mechanik bereitzustellen. Dabei wird die Vision verfolgt, die geforderte/gewünschte Funktionsweise einer noch nicht existierenden Software auf noch nicht existierender Hardware (Elektronik / Mechanik) abzubilden, die Gesamtfunktionalität zu simulieren und damit digitale (Geometrie-) Modelle zu steuern.

Das FDMU-Rahmenwerk wurde anhand dreier Anwendungsszenarien evaluiert. Im vorliegenden Artikel werden unser methodisches Vorgehen und die system-technische Umsetzung beschrieben. Weiterhin wird ein ausgewähltes Anwendungsszenario beleuchtet.

Herausforderungen ‚Heterogenität‘ und Lösungsansatz

Wie bereits erwähnt, wird in den unterschiedlichen mechatronischen Domänen eine Vielzahl von unterschiedlichen Systemen eingesetzt. Diese Systeme basieren auf verschiedenen Verhaltensbeschreibungs- bzw. -modellierungssprachen. Diese Verhaltensmodelle werden in dedizierten Simulatoren ausgeführt. Zwar existieren für bestimmte Simulatoren Möglichkeiten des Modellexports bzw. -austauschs, jedoch ist ein solcher Ansatz als generische Lösung kaum praktikabel. Aus diesem Grund verfolgen wir als Lösungsansatz eine verteilte, herstellerunabhängige Co-Simulationsumgebung mit erweiterter 3D-DMU-Visualisierungskomponente, die als einheitliche Benutzerschnittstelle zur Systemsimulation dient. Abbildung 1 zeigt eine beispielhafte Ausprägung der FDMU-Laufzeitumgebung sowie die Informationen, die den einzelnen Komponenten zur Verfügung gestellt werden.

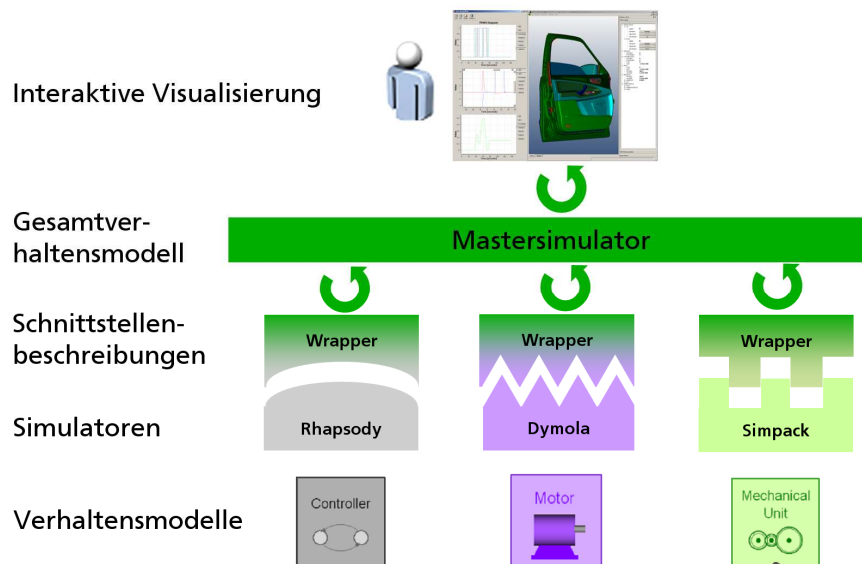


Abbildung 1: Beispielhafte Ausprägung der FDMU-Laufzeitumgebung

Der Aufbau der FDMU-Laufzeitumgebung von unten nach oben:

- Native Verhaltensmodelle, die durch
- entsprechende Simulatoren ausgeführt werden.
- Wrapper, die die unterschiedlichen Schnittstellen der Simulatoren ‚nach oben hin‘ vereinheitlichen und an einen Mastersimulator anbinden. Die Wrapper erhalten zur Laufzeit eine Beschreibung der Schnittstellengrößen der Verhaltensmodelle und bilden ein- und ausgehende Signale auf diese ab bzw. wandeln ausgehende Größen ggfs. um. Weiterhin steuern die Wrapper den gekapselten Simulator an.
- Dem Mastersimulator sind die internen Schnittstellen des Gesamtverhaltensmodells bekannt. Er ist für die Kommunikation und Koordination der Simulatoren zuständig.
- Die interaktive Visualisierung erlaubt das Ausführen und Parametrisieren von Gesamtsimulationen.

Wie werden nun native Verhaltensmodelle in eine Gesamtsimulation eingebunden?

Der Ansatz FDMU geht davon aus, dass Verhaltensmodelle in unterschiedlichen Modellierungssprachen vorliegen. Diese Modellierungssprachen verwenden u.U. unterschiedliche Syntax, Einheitensysteme, Solver, etc. Ebenso kann die Namensgebung der Schnittstellengrößen der Einzelmodelle voneinander abweichen – Verhaltensmodelle könnten beispielsweise von einem Zulieferer entwickelt und beigestellt worden sein. Dies macht es erforderlich, die Schnittstellengrößen aufeinander abzustimmen.

Als universelle Schnittstellenbeschreibungssprache haben wir in FDMU SysML eingesetzt. SysML ist international standardisiert und wird als XML-basierte Sprache von einer zunehmenden Anzahl von XML-Werkzeugen unterstützt. SysML erlaubt es, die Schnittstellen der nativen Verhaltensmodelle standardisiert und werkzeugunabhängig zu beschreiben.

Das Beschreiben der Schnittstellen der nativen Verhaltensmodelle erfolgt graphisch-interaktiv durch den Benutzer mit Hilfe des FDMU-Editors. Der Benutzer führt alle benötigten nativen Schnittstellengrößen nach ‚außen‘ und bildet diese auf standardisierte Typen in SysML ab. Er kann den Verhaltensmodellen entsprechende Geometriemodelle zuordnen, so dass FunctionalBuildingBlocks (FBB) entstehen, die Geometrie und Funktion nach einem objekt-orientierten Vorbild kapseln. Z.B. ordnet er einem FBB ‚Elektromotor‘ sein elektro-mechanisches Verhaltensmodell und ein entsprechendes Geometriemodell zu. Durch Verbinden mehrerer mit SysML beschriebener Verhaltensmodelle entsteht ein Systemmodell. Abbildung 2 zeigt beispielhaft einen Ausschnitt aus einem einfachen Systemmodell für einen Fensterheber:

- die nativen Verhaltensmodelle (Controller, Motor, Mechanical Unit) sind jeweils in einer ‚SysML-Schale‘ gekapselt,
- wichtige interne Schnittstellengrößen der einzelnen Verhaltensmodelle sind auf die SysML-Ebene herausgeführt,
- entsprechende Geometriemodelle sind zugeordnet.

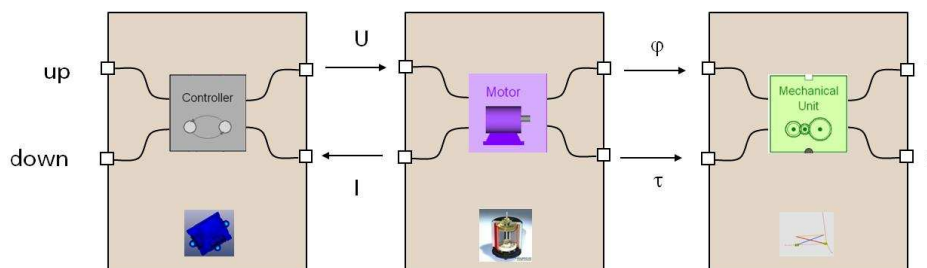


Abbildung 2: Ausschnitt aus einem einfachen Systemmodell für einen Fensterheber

Anzumerken bleibt, dass die nativen Verhaltensmodelle für die Größen, die herausgeführt werden sollen mit sog. Konnektoren angereichert werden müssen – zu diesem Zweck ist während des Projektes FDMU eine Bibliothek entstanden.

Der Aufbau des Gesamtsimulationsmodells in SysML erlaubt die Konsistenzprüfung des Modells auf abstrakter Ebene. Darüber hinaus können zu den Parametern erlaubte Wertebereiche angegeben werden, die während des Simulationslaufs geprüft werden können.

Wie werden unterschiedliche Simulatoren in das FDMU-Rahmensystem eingebunden?

Der Variantenreichtum an Simulatoren ist ebenso groß wie der an Verhaltensmodellierungssprachen. Die Simulatoren unterscheiden sich hinsichtlich ihrer Programmierschnittstellen und der Art und Weise wie sie in ‚fremde‘ Umgebungen eingebunden werden können. Zur Einbindung in das FDMU-Rahmenwerk werden Wrapper verwendet, die die individuellen Programmierschnittstellen der Simulatoren auf eine einheitliche Schnittstelle abbilden. Dazu wird der Mechanismus zur Einbindung externer Funktionen verwendet, den viele Simulatoren bereitstellen.

Die Wrapper übernehmen zwei Aufgaben:

- Sie steuern den darunterliegenden Simulator, d.h. sie stellen in der Initialisierungsphase der Gesamtsimulation die Verhaltensmodelle bereit, starten die Simulation, halten die Simulation ggfs. an, setzen sie fort oder beenden sie – alles auf Kommando des Benutzers oder einer Kontrolldatei.
- Sie stellen die Kommunikation mit dem Mastersimulator sicher, nehmen Eingangsdaten entgegen und geben Ergebnisse zurück. Dabei bilden sie die nativen Schnittstellengrößen auf die standardisierten ab und umgekehrt.

Wie kommunizieren die Simulatoren im FDMU-Rahmensystem?

Der Mastersimulator ist Dreh- und Angelpunkt der Kommunikation und Synchronisation im FDMU-Rahmensystem. Bei dem Mastersimulator handelt es sich um eine Entwicklung aus dem FDMU-Projekt. Der Mastersimulator koordiniert die Ausführung des Gesamtsimulationsmodells. Er überwacht mit Hilfe der Wrapper die Einzelsimulationen und stellt die Weitergabe der Daten zwischen den Wrappern und anderen angeschlossenen Diensten, wie z.B. der FDMU-Visualisierung, sicher. Der Mastersimulator ist der Taktgeber für die Gesamtsimulation, er enthält selbst keinen Simulator. Vorbereitet sind Schnittstellen, die die Integration unterschiedlicher Mastersimulator-Algorithmen erlauben, über die auf die Reihenfolge der Einzelsimulationen Einfluss genommen werden kann. Ziel ist hierbei, Schrittweiten dynamisch anzupassen und – so es die Einzelsimulatoren erlauben – bei Notwendigkeit zu einem Zeitpunkt eine Lösung iterativ zu bestimmen.

Eine wichtige Komponente des Mastersimulators ist der TransferHandler, er sorgt für eine angepasste Datenkommunikation zwischen den angebundenen Simulatoren. Abbildung 3 zeigt ein Schema, bei dem der Benutzer zu beliebigen Zeitpunkten einen Schalter (hier Fensterheber) bedient, der Controller allerdings ein Signal mit äquidistanten Stützstellen erwartet. Der TransferHandler kann so parametrisiert werden, dass er die Signale automatisch anpasst. Er verfügt über weitere Protokolle wie upsampling, downsampling, etc.

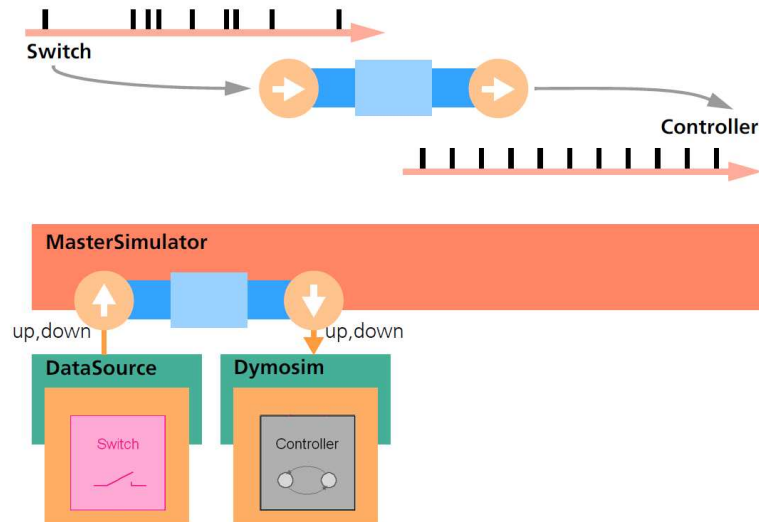


Abbildung 3: Beispiel für den TransferHandler im Mastersimulator

Wie interagiert der Benutzer mit dem FDMU-Rahmensystem?

DMU stellt generell hohe Anforderungen an die Interaktivität mit großen Datenmengen, klassischerweise von geometrischer Art. Daher liegt es nahe, auch für FDMU Interaktivität zu fordern, was gesteigerte Anforderungen an die Simulatoren stellt, da diese typischerweise nicht für interaktive Simulationen prädestiniert sind. Interaktive Simulationen sind dadurch gekennzeichnet, dass vor Beginn der Simulation nicht feststeht, welches Ereignis wann eintritt. Jedoch erfordern nicht alle Anwendungsszenarien von FDMU Interaktivität. Es ist auch vorstellbar, FDMU für das automatisierte Abarbeiten vordefinierter Testszenarien einzusetzen. Zu diesem Zweck wurde alternativ die Verarbeitung von Eingabedateien vorgesehen.

Die FDMU-Visualisierung ermöglicht dem Benutzer das interaktive Explorieren von Gesamtsimulationen. Dies umfasst:

- eine 3-dimensionale Darstellung und Navigation der (DMU-)Geometriedaten,
- das Erzeugen von Simulationsereignissen in der 3D-Szene oder mit Hilfe von Buttons,
- das Ändern von Simulationsparametern, die unmittelbar an die Simulatoren weitergegeben werden,
- das Beobachten von Simulationsverläufen in der 3D-Szene und
- das Analysieren von Simulationsgrößen in Form von Graphen.

Dabei werden alle ausgewählten Daten, die über den Mastersimulator kommuniziert werden, in einer vereinheitlichten Benutzungsschnittstelle dargestellt.

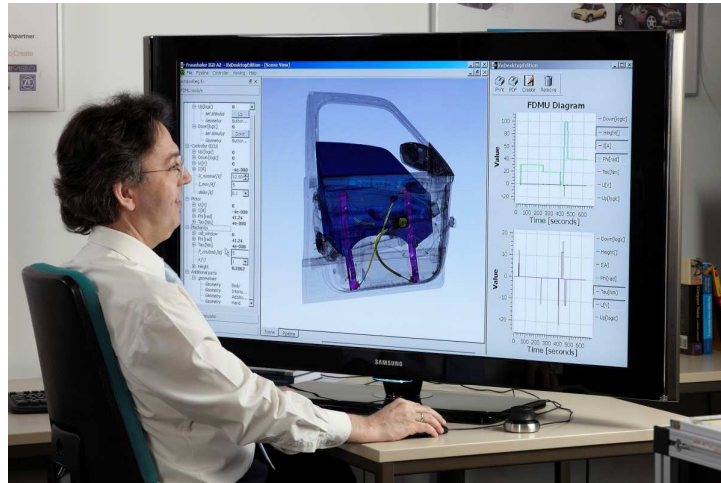


Abbildung 4: FDMU-Visualisierung

Abbildung 4 zeigt die FDMU-Visualisierung mit der Darstellung der Geometriedaten, einer Navigations- und Interaktionsleiste auf der rechten Seite sowie auf der linken Seite die Möglichkeit der graphischen Präsentation von Signalverläufen. Es können mehrere Plots kombiniert werden, die dargestellten Simulationsgrößen sind jeweils frei wählbar. Alle Größen werden auf einer gemeinsamen Zeitachse dargestellt.

In der Mitte befindet sich der 3D-Graphikbereich, der die Darstellung auch großer DMU-Modelle erlaubt. Hier können auch Bewegungen dargestellt und beobachtet werden, die durch die Gesamtsystemsimulation induziert sind. Weitere Funktionalitäten sind das Darstellen von Signalverläufen (Quelle/Senke), das Hervorheben von Geometrien, die Kollisionserkennung, etc.

Auf der rechten Seite finden sich die Repräsentationen der FBB. Hier lassen sich Parameterwerte setzen, ändern, aktuelle Simulationsgrößen ablesen und dergleichen mehr. Die Bedienelemente für die FBB werden automatisch aus der Gesamtsystembeschreibung generiert, inkl. Randbedingungen für die Parameter wie sie vorher spezifiziert wurden, z.B. diskrete Werte oder erlaubte Wertebereiche.

Ausgewähltes Anwendungsszenario: thermodynamische Simulation E-Motor und Regler

Ein Elektromotor ist mit einer Ansteuerelektronik inklusive Sollwertgeber versehen. Die Elektronik befindet sich auf einer Leiterplatte, die außen an der Stirnseite des Motors angebracht ist. Das elektrische Verhalten der Bauelemente auf der Leiterplatte wird aufgrund der Wärmeentwicklung beim Betrieb des Motors thermisch beeinflusst.

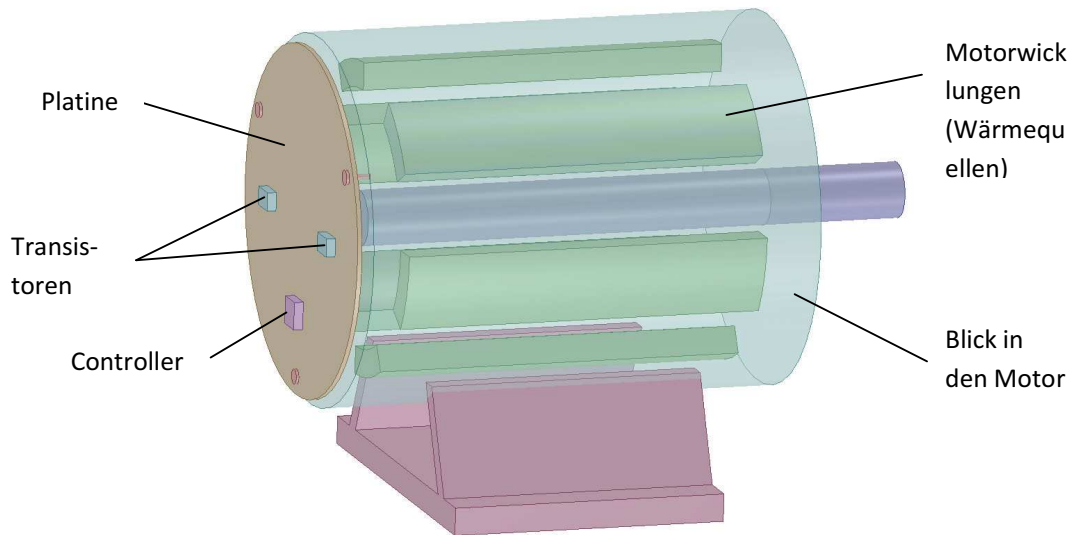


Abbildung 5: E-Motor mit Regler an der Motorenrückseite

Typische Entwurfsfragestellungen, die beantwortet werden sollen, lauten:

- Ist der Motor richtig ausgelegt?
- Wie hoch ist die Erwärmung der Leistungstransistoren?
- Wie groß ist der zusätzliche Einfluss der Motorerwärmung auf die Transistoren?
- Hat die Motorerwärmung Einfluss auf andere Bauelemente (z.B. Controller)?
- Was passiert, wenn bei den Transistoren größere Kühlkörper verwendet werden?
- Hilft ein größerer Abstand zwischen Motor und Platine?

Das Modell wurde wie in Abbildung 6 dargestellt zerlegt. Die elektrische Einheit umfasst die Leistungsverstärker. Die mechanische Einheit enthält ein Dämpfungselement und übersetzt die Lastprofile auf den Motor. Elektrische Einheit und Motor wechselwirken sowohl elektrisch als auch thermisch miteinander. Die Schnittstellengrößen sind in Abbildung 6 ebenfalls genannt.

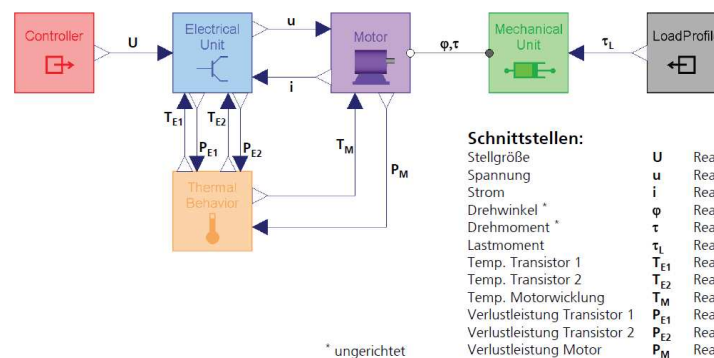


Abbildung 6: Zerlegung des Gesamtsystems

Dieses Anwendungsszenario bindet eine Thermodynamik-Simulation basierend auf der FE-Methode mit in die Gesamtsimulation ein und unterstreicht das potenziell weite Einsatzgebiet des FDMU-Rahmenwerkes. Das Wärmeausbreitungsmodell wurde mit Hilfe von Ansys auf einem Netz mit 50.000 Knoten realisiert. Um die Berechnungszeiten zu verkürzen und die Simulation interaktiver zu

gestalten, wurden Modellordnungsreduktionsverfahren eingesetzt. Damit wurden testweise zwei Modelle (eines mit 100 Knoten, eines mit 7 Knoten) erzeugt. Die drastische Reduktion der Anzahl der Modellgleichungen ergab eine deutliche Verbesserung der Performanz des Gesamtsystems bei Erhaltung der Haupteigenschaften des thermodynamischen Teilmodells.

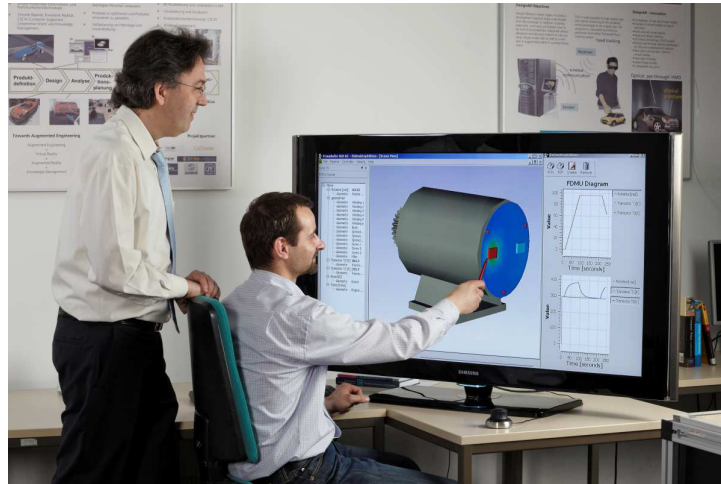


Abbildung 7: Visualisierung des Ergebnisses in FDMU-Visualisierung

In Abbildung 7 ist ein Momentanbild einer möglichen Visualisierung der thermischen Ergebnisse dargestellt. Die aktuellen Temperaturen der Messpunkte werden als Farbwert auf den Knoten eines Dreiecksnetzes dargestellt. Dazwischen erfolgt eine Interpolation. Somit kann die Temperaturverteilung während der Simulation bequem verfolgt werden.

Das Beispiel des thermodynamischen E-Motors zeigt eindrucksvoll die Palette der Kopplungsmöglichkeiten, die das FDMU-Rahmensystem derzeit beherrscht. Voll ausgebaute Lösungen existieren bereits für Simulatoren wie Dymola, Saber, Simpack und Matlab/Simulink, mit denen u.a. die Verhaltensbeschreibungssprachen Modelica, MAST, VHDL-AMS sowie Simulink-S-Functions abgedeckt werden. Für FE-Modelle besteht aktuell eine prototypische Lösung, die in Zukunft weiter ausgebaut wird.

Nutzen und Vorteile von FunctionalDMU

Mit FunctionalDMU wurde ein neues Design-Review-Werkzeug mit integrierter Darstellung und Beeinflussung von multi-domänen Simulationen geschaffen. Eine einheitliche Benutzungsschnittstelle ermöglicht einen zentralen Zugang zu gekoppelten Simulationen. Die Visualisierungsmöglichkeiten unterstützen die Kommunikation von Lösungsvorschlägen und auftretenden Problemen im Gesamtsystem über Domänengrenzen hinweg.

Das FDMU-Rahmenwerk ermöglicht einen schnellen Aufbau von gekoppelten Simulationen, ohne dass der Anwender sich um spezifische Simulatorkopplungen kümmern muss. Die flexible Komposition von Simulatoren und Verhaltensmodellen ermöglicht generell neue Möglichkeiten zur gekoppelten multi-domänen Simulation.

Probleme im Zusammenspiel von Verhaltensmodellen können frühzeitig erkannt werden und damit können Rückschlüsse auf das zu erwartende Systemverhalten gezogen werden. Die Auslegung von Gesamtsystemen, oder die von Teilsystemen, kann im Kontext erfolgen. Konsistenzprüfungen können auf Modellebene erfolgen, mögliche Konflikte zwischen den Verhaltensmodellen der Teilsysteme bzw. bezüglich der Annahmen über Ein- und Ausgangsparameter können während der Simulation aufgedeckt werden. Zukünftig ist eine Einbettung von FDMU in Optimierungsansätze denkbar.

Die Verteilbarkeit des Systems trägt Anforderungen nach Nutzung existierender Ressourcen Rechnung, insbesondere von Lizenzen und Rechnerkapazität.

Zusammenfassung

Mit dem FunctionalDMU-Framework ist die simulative Kopplung der mechatronischen Domänen über die Grenzen kommerzieller Simulationswerkzeuge hinaus möglich. Auf diese Weise lassen sich die Entwicklungszeiten mechatronischer Systeme verkürzen und funktionales Design Review wird aufwandsmäßig besser beherrschbar.

Das FunctionalDMU-Framework baut auf hersteller- und plattformunabhängigen Technologien auf und kann somit in unterschiedlichen Umgebungen eingesetzt werden. Des Weiteren zeichnet sich das Framework durch seine offene und leicht erweiterbare Architektur aus. So können neben den bereits existierenden Anbindungen der Simulatoren Rhapsody, SimPack, Saber, Dymola (Modelica), Matlab/Simulink, auch weitere Simulatoren angebinden werden. Durch den Einsatz einer service-orientierten Architektur können die Rechenlasten auf unterschiedliche Systeme verteilt werden.

Für die strukturierte Einführung und den Einsatz von FDMU wurde eine Modellierungs-, Integrations- und Nutzungsmethodik entwickelt. Diese erlauben dem Benutzer z.B. eine flexible Nutzung von Simulatoren und Hardware, eine schnelle Beschreibung des Gesamtverhaltens, sowie ein Erkennen von Fehlern in früheren Phasen der Produktentwicklung.

Die Umsetzbarkeit domänenübergreifender Simulationen wurde anhand des beschriebenen Szenarios plus zweier weiterer Szenarien (elektrischer Fensterheber und Lenkungsprüfstand) nachgewiesen.