

# Werkzeuge für die Entwicklung mechatronischer Systeme mit Methoden des MBSE

Alexander Blumör<sup>1</sup>, Gerhard Pregitzer<sup>1</sup> und Martin Bothen<sup>2</sup>

<sup>1</sup>Karl Mayer Textilmaschinenfabrik GmbH, Brühlstraße 25, 63179 Obertshausen,  
alexander.blumoer@karlmayer.com

<sup>2</sup>Hochschule Aschaffenburg, Würzburger Straße 45, 63743 Aschaffenburg,  
martin.bothen@h-ab.de

**Zusammenfassung:** Die Entwicklungsmethodik für mechatronische Systeme benötigt eine sehr gute Integrationsfähigkeit der verwendeten (Software-) Werkzeuge für den Systementwurf, den domänenspezifischen Entwurf, sowie für die Systemintegration. Hierzu wurden verschiedene Werkzeuge untersucht, die eine Abbildung des Model Based Systems Engineering (MBSE) ermöglichen. Insbesondere Werkzeugumgebungen für integriertes und föderiertes MBSE wurden untersucht. Ein Hauptaugenmerk wurde auf die Integration der Simulation in Verknüpfung mit SysML oder ähnlichen Modellen gelegt. Zu diesem Zweck wurde ein Referenzsystem mit verschiedenen Werkzeugen virtuell aufgebaut und der Umfang der Werkzeuge bewertet. Zum Aufbau des Referenzsystems wurde grundlegend mit dem RFLP-Ansatz und der FAS-Methode vorgegangen. Ferner wurde untersucht, ob die vorliegenden Methoden ausreichend sind, um MBSE mit den Werkzeugen zu betreiben. Die Arbeit zeigt, dass die Methoden im Umfeld des MBSE noch weiter ausgearbeitet werden müssen.

## 1 Einleitung

Es existieren verschiedene Werkzeuge, die MBSE unterstützen, um damit komplexe Systeme zu entwickeln. Grundlegend sind die MBSE-Ansätze in integrierte und föderierte Werkzeuglösungen zu unterscheiden. Für Anwender dieser Werkzeuge, wie KARL MAYER oder die Hochschule Aschaffenburg, stellt sich hierbei die Frage, welche Werkzeuge am besten für den eigenen Einsatzfall geeignet sind und wie aufwändig sich ein Vorgehen darstellt. Historisch bedingte Strukturen, wie vorhandene Werkzeuge (CAx-Systeme), Datenablagensysteme (PLM-Systeme), Abbildung von MBSE-Methoden sowie die Durchgängigkeit von Daten sind die Bewertungsgrößen, an denen die Werkzeuge für integriertes und föderiertes MBSE gemessen werden. Untersucht wird, wie gut MBSE mit existierenden Werkzeugen und Methoden technisch umsetzbar ist. Dabei wird werkzeugseitig der integrierte Ansatz mit dem föderierten Ansatz verglichen.

Die verwendeten Methoden werden in Kapitel 2 beschrieben. Kapitel 4 beschreibt das Anwendungsprojekt und dessen genaue Ausprägung bei den verwendeten Werkzeugen. Aus dem Anwendungsprojekt werden dann die Ergebnisse abgeleitet.

## 2 Methodische Ansätze für MBSE

Dieses Kapitel zeigt die Ansätze für föderiertes und integriertes MBSE auf. Am Ende des Kapitels werden die verwendeten MBSE-Methoden näher erklärt, mit denen integrierter und föderierte MBSE-Ansatz durchgeführt werden können. Dabei wird insbesondere auf die FAS-Methode und den RFLP-Ansatz wert gelegt.

### 2.1 Föderiertes Model Based Systems Engineering

Föderiertes MBSE beschreibt die Verwendung von verschiedenen, aber für den jeweiligen Anwendungsfall, optimalen Werkzeuge mit Methoden des MSBE. Dabei werden die Werkzeuge durch Schnittstellen und Austauschformate modellbasiert verknüpft.

Der Vorteil des föderierten MBSE kommt aus den Werkzeugen selbst. So können für einzelne Teilaufgaben des MBSE jeweils die optimalen Werkzeuge eingesetzt werden. Für die Erstellung des Systemmodells kann dadurch der passende Modellierer und für die mechatronische Systemsimulation der passende Simulator verwendet werden. Im Unternehmenskontext bedeutet dies, dass der Mehrwert des MBSE mit bestehenden Werkzeugen und Modellen erreicht werden kann, ohne eine umfassende Änderung in der Werkzeuglandschaft vornehmen zu müssen. Die Vorgehensweise setzt aber auch Integrator-Werkzeuge, ein eigenes Datenmodell und eine versionsgerechte Werkzeugverknüpfung voraus. Selbst unter der Verwendung von definierten Schnittstellen bleibt die Schnittstellenkomplexität beim föderierten MBSE hoch. Diese Komplexität muss dann vom Anwender oder der IT-Abteilung beherrscht werden. Die Werkzeugketten sind hier aufs höchste Maß auf die jeweiligen Unternehmen zugeschnitten. [KI03]

### 2.2 Integriertes Model Based Systems Engineering

Integriertes MBSE beschreibt die Verwendung einer integrierten Werkzeugumgebung für verschiedene Anwendungsfälle des MBSE. Die Werkzeugumgebung stammt dabei von einem einzigen Hersteller. Dadurch entfällt die Schnittstellenproblematik aus 2.1, da die Modelle über das Datenmodell des Werkzeugherstellers verknüpft sind. So werden Applikationen für Teilaufgaben des MBSE bereitgestellt und in einem Datenmodell integriert. In bestimmten Anwendungsfällen hat dies aber auch zur Folge, dass eben nicht das optimale Werkzeug eingesetzt werden kann und immer eine gewisse Abhängigkeit zum Hersteller besteht.

Der Anwender arbeitet in einer einzigen Entwicklungsumgebung und sollte sich nicht mit Schnittstellen, Austauschformaten oder Datenmodellen befassen müssen. Die integrierte Werkzeugumgebung muss also die Funktionen abbilden, mit denen gearbeitet werden soll. Die Einführung einer integrierten Werkzeuglösung ist immer mit einem recht großen strukturellen Wandel verbunden, wenn vorher föderiert gearbeitet wurde. Vorhandene Werkzeuge müssen durch die integrierte Werkzeugkette substituiert und Alt-Daten migriert werden. [KK13]

Hersteller wie *Dassault Systèmes* oder *Siemens* bieten fertige Produktportfolios für integriertes MBSE an, die vom Anforderungsmanagement bis zum Betrieb beim Kunden alle Themenfelder des MBSE abdecken.

### 2.3 FAS-Methode und RFLP-Ansatz

Die VDI-Richtlinie 2206 beschreibt die Entwicklungsmethodik für mechatronische Systeme von den Anforderungen über den Systementwurf, dem domänenspezifischen Entwurf, der Systemintegration hin zum fertigen Produkt [VDI04]. Ein wichtiger Aspekt bei dieser Entwicklungsmethodik ist die Modellbildung und Modellanalyse, die den Entwicklungsprozess kontinuierlich begleitet. Der in Bild 1 dargestellte RFLP-Ansatz fordert zusätzlich noch ein Datenmodell, das die Systemmodelle beinhaltet und zu jedem Zeitpunkt im Entwicklungsprozess und Lebenszyklus die gezeigten Daten bereitstellt.

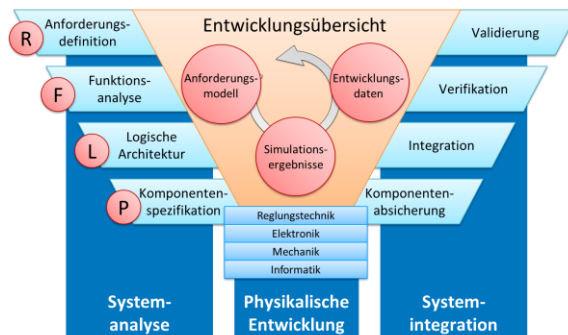


Bild 1. RFLP-Ansatz [B116], [KK13]

Der RFLP-Ansatz ist ein methodischer Ansatz, der die FAS-Methode unterstützt und ergänzt. Er gibt die Entwicklungsschritte einer Systementwicklung vor und detailliert diese. Der RFLP-Ansatz ergänzt die VDI 2206. [KK13], [Wt16]

In vorangegangenen MBSE-Projekten ([KK13], [PBK14]) wurde auf die FAS-Methode und den RFLP-Ansatz gesetzt, aber der Vergleich zwischen integriertem und föderiertem Ansatz wurde noch nicht durchgeführt. Andere Methoden, wie bspw. SYSMOD erschienen für ein Maschinenbauunternehmen wie KARL MAYER eher ungeeignet, da die FAS-Methode explizit die funktionale Sicht auf Systeme fordert, wie diese zum Beispiel auch häufig im Maschinenbau angewendet wird. Der RFLP-Ansatz deckt zusätzlich auch andere Bereiche, von der Systemmodellierung, über Simulation bis zur Komponentenspezifikation mit ab.

Die FAS-Methode beruht auf die weitverbreitete funktionale Analyse von Systemen. Über verschiedene Diagramme wird das System beschrieben, siehe Bild 2. Dabei wird aus den Anwendungsfällen, Anforderungen und dem System Kontext eine funktionale Architektur aufgebaut. Mit Kreativmethoden werden dann aus den Lösungsprinzipien Konzepte abgeleitet.

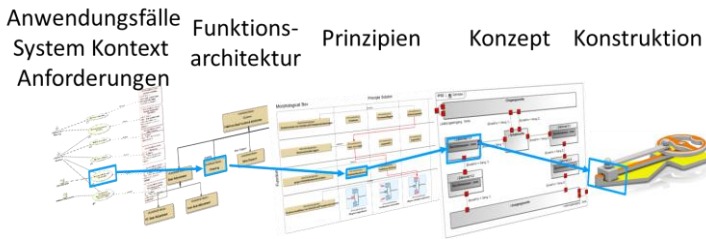


Bild 2. Vorgehensweise bei der FAS-Methode [BPK15]

Zwar wurden FAS-Methode und RFLP-Ansatz in den vorangegangenen Projekten mit einer föderierten Werkzeugumgebung getestet, aber noch nicht mit einem integrierten System benutzt. Um föderiertes und integriertes MBSE zu vergleichen, wird für diese Aufgabe die direkte Gegenüberstellung mit FAS-Methode und RLFP-Ansatz erstmalig durchgeführt.

### 3 Anwendungsprojekt

Im Anwendungsprojekt wird ein mechatronisches, ferngesteuertes Fahrzeug, das RECT (Remote Controlled Energy Car T), aufgebaut, um integrierten und föderierten Ansatz zu vergleichen. Dabei werden die Schritte des RFLP-Ansatzes und der FAS-Methode in den jeweiligen Werkzeugumgebungen abgebildet. Dieses Kapitel erläutert die im Detail eingesetzten Werkzeuge, begründet deren Auswahl und gibt einen Einblick in den Verlauf des Anwendungsprojekts.

#### 3.1 Werkzeuglandschaft

KARL MAYER setzt *CATIA V5* von *Dassault Systèmes* als Konzern-CAD-System mit *SAP-PLM* ein. Dieses CAD-Programm wird ebenfalls für die Konstruktionsausbildung im Studiengang Mechatronik der Hochschule Aschaffenburg verwendet. Als Nachfolgeprodukt von *CATIA V5* wird von *Dassault Systèmes* die *3DEXperience* Plattform angeboten. Die *3DEXperience* Plattform wird häufig auch als *CATIA V6* bezeichnet. Es ist ein integriertes Werkzeug, das viele Entwicklungswerkzeuge vereint, weswegen es im Anwendungsprojekt untersucht werden soll.

Die föderierte Lösung wird mit den bei KARL MAYER vorhandenen Werkzeugen *ModelCenter (Phoenix Integration)*, *SimulationX (ESI ITI)*, *CATIA V5* und der *Cameo Systems Modeler (No Magic)* aufgebaut. Das Systemmodell wird im *Cameo Systems Modeler (CSM)* benutzt, um alle Daten des Entwicklungsprozesses an einer Stelle abzubilden. *SimulationX* wird für die Konzeptsimulation und weiter bis zur Systemsimulation und virtueller Inbetriebnahme verwendet. Bild 3 stellt ein Schema der verwendeten Werkzeuge im föderierten MBSE dar. Als CAD-System kommt *CATIA V5* zum Einsatz. Diese Werkzeugumgebung hat sich in vorangegangenen Projekten als geeignet für föderiertes MBSE erwiesen ([BPK15], [PBK14]).

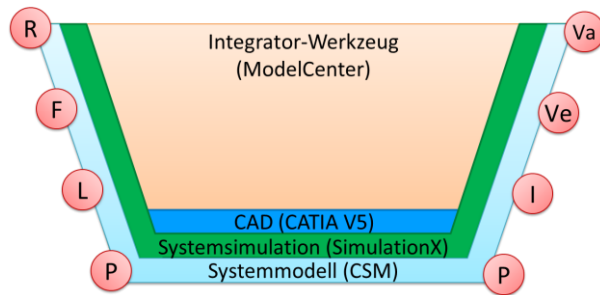


Bild 3. Föderierte Werkzeuglandschaft

Die 3DExperience Plattform 2016x arbeitet für die verschiedenen Aufgabenbereiche des MBSE mit sogenannten Apps. Welche Apps verwendet werden, ist in Bild 4 dargestellt. Dabei ist anzumerken, dass *Enovia* als PLM-System der 3DExperience Plattform das Datenmodell und somit das Systemmodell verwaltet und speichert. In der App *Requirements* wird das Anforderungsmanagement betrieben. *Functional Logical Design* ist der Editor für die funktionale Architektur, Konzeptsimulation und Systemsimulation. Die App *CATIA V6* ist stark in die *Functional Logical Design* App integriert und dient als mechanisches CAD. In der *Behavior Experience* App lassen sich Systeme bis zur virtuellen Inbetriebnahme abbilden. Dafür werden die Modelle aus dem *Functional Logical Design* weiterverwendet.

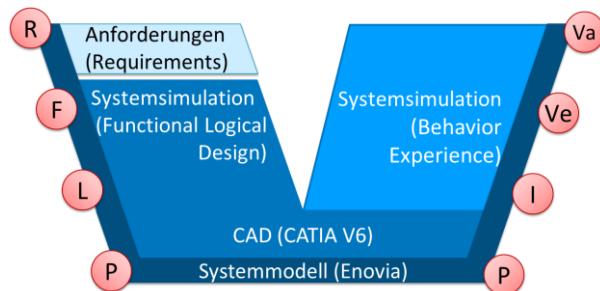


Bild 4. Integrierte Werkzeuglandschaft mit 3DExperience

### 3.2 MBSE in föderierter Werkzeugumgebung

Im *CSM* findet die gesamte Modellierung des Systemmodells in SysML ohne Schnittstellen in einem Werkzeug statt. Dadurch, dass die FAS-Methode auch als SysML-Methode mit diesem Werkzeug entwickelt wurde, ist die Durchgängigkeit von Anforderung bis zur Komponentenspezifikation problemlos gegeben.

Das SysML-Modell beinhaltet im *CSM* das Anforderungsmodell, den System Kontext und die Anwendungsfälle für das RECT. Das Anforderungsmodell unterteilt sich in funktionale und nichtfunktionale Anforderungen, wobei nichtfunktionale Anforderungen

mit Performance-Parametern belegt werden können. Diese Parameter dienen bei der Anforderungvalidierung als Abfrageelemente gegen Simulations- oder Testergebnisse.

Aus den Anwendungsfällen, dem System Kontext und den Anforderungen wird die funktionale Architektur des RECT erstellt, siehe Bild 5. In der funktionalen Architektur des SysML-Modells finden sich alle geforderten Funktionen des RECT wieder.

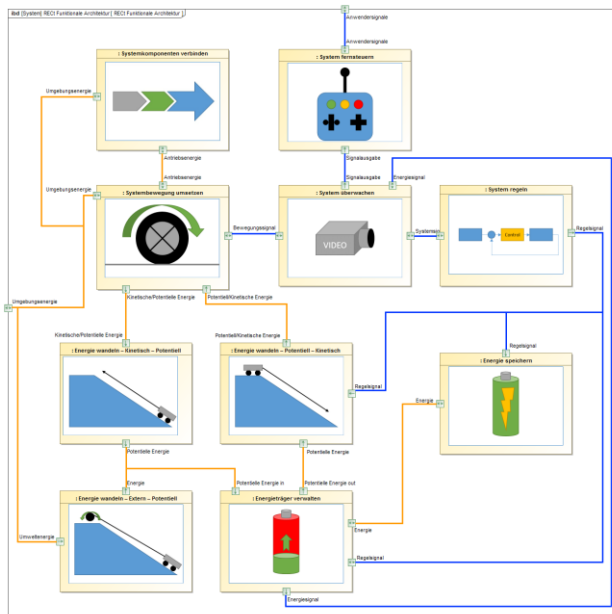


Bild 5. Funktionale Architektur des RECT in SysML [BI16]

Im morphologischen Kasten wird mit SysML dokumentiert, wie die funktionalen Elemente des Systems realisiert werden sollen. Aus dem morphologischen Kasten und der funktionalen Architektur werden Konzepte als logische Architektur ausgearbeitet. Dabei werden, wie Bild 6 zeigt, die Komponenten und deren Schnittstellen definiert. Aus den logischen Architekturen der Konzepte müssen nun Konzeptsimulationsmodelle abgeleitet werden. Zum Zeitpunkt dieser Arbeit gab es noch keine Möglichkeit dies automatisiert durchzuführen. Zunächst entsteht zwischen logischer Architektur im *CSM* und dem Simulationsmodell in *SimulationX* ein Medienbruch in der Struktur der beiden Modelle. Dieser wird während der Entwicklung manuell nachgepflegt. Eine automatische Ableitung von Simulationsmodell aus SysML-Modell ist momentan nicht möglich.

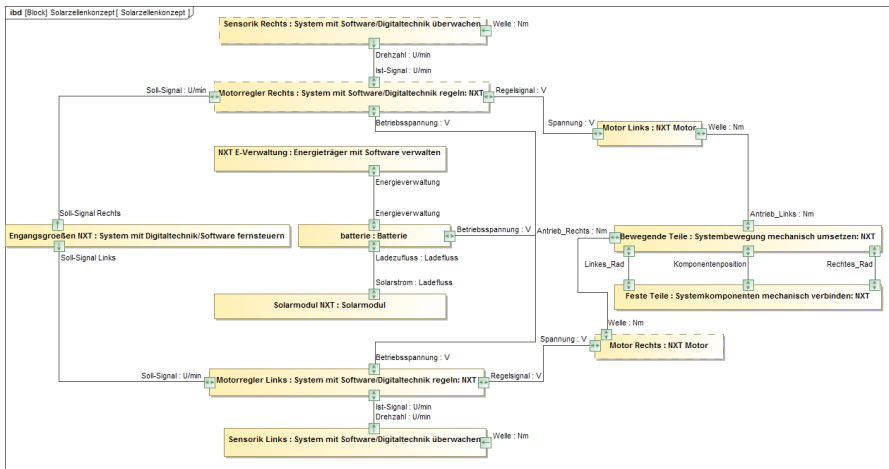


Bild 6. Logische Architektur der beiden Konzepte [B116]

Bild 7 zeigt das fertige Simulationsmodell, das aus der logischen Architektur abgeleitet wurde. Auf Basis der logischen Architektur wird ein *CATIA-V5-CAD-Modell* aufgebaut und manuell in das Simulationsmodell importiert. Dies stellt den zweiten, strukturellen Medienbruch dar. Änderungen im CAD-Modell müssen aus *CATIA V5* manuell ins Simulationsmodell importiert werden.

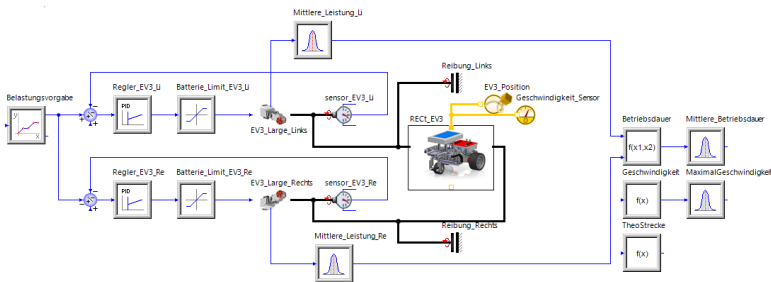


Bild 7. Simulationsmodell [B116]

Nach der Vorbereitung des SysML-Modells und einer Integration von Systemmodell, Simulationsmodell und CAD-Modell in *ModelCenter* können die Partialmodelle parametrisch verknüpft werden, siehe Bild 8. Es ist nicht möglich die Struktur automatisiert zu synchronisieren, wohl aber die Parameter. Ändert sich die Struktur in den Teilmodellen, muss diese über alle anderen Modelle hinweg gleichgezogen werden. Durch die Verknüpfung in *ModelCenter* ist mit Parameterstudien das automatisierte Abfragen von Anforderungen mit Anforderungssampeln für alle Systemanforderungen möglich.

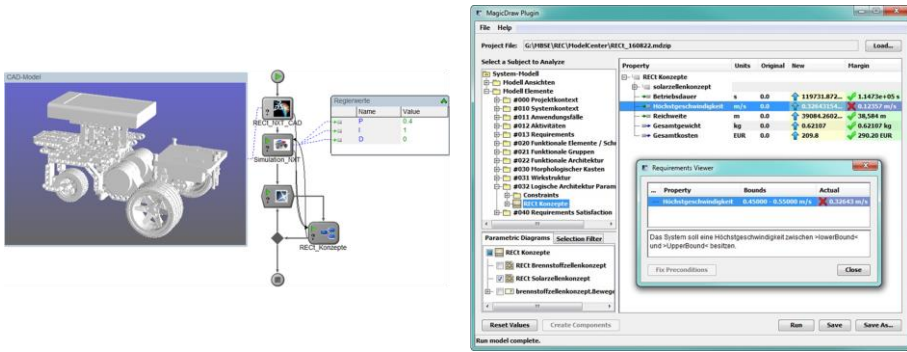


Bild 8. ModelCenter-Modell mit Anforderungsampel [B116]

### 3.3 MBSE in integrierter Werkzeugumgebung

In der integrierten Werkzeugumgebung werden die Anforderungen an das RECT in der *Requirements*-App angelegt und gesammelt. Dabei handelt es sich um ein reines Anforderungsmanagement-Werkzeug, in dem die Anforderungen als Liste mit Eigenschaften aufgenommen werden. Die Anforderungen besitzen keine Validierungsparameter. Die Darstellungen aus der FAS-Methode lassen sich nicht in dieser App implementieren. Hierfür muss in die App *Functional Logical Design* gewechselt werden. Die *3DExperience* Plattform setzt nicht auf SysML als Modellierungssprache, sondern auf eine eigene funktionale Beschreibungsnotation ein, die nicht die Freiheiten von SysML bietet. Dadurch verlieren die Diagramme für System Kontext oder Anwendungsfälle die Verknüpfung zu den anderen Modellobjekten und werden so zu Bildern, ohne eine modellbasierte Verknüpfung. Bild 9 zeigt, dass die reine Abbildung der funktionalen Architektur des RECT jedoch gut möglich ist.

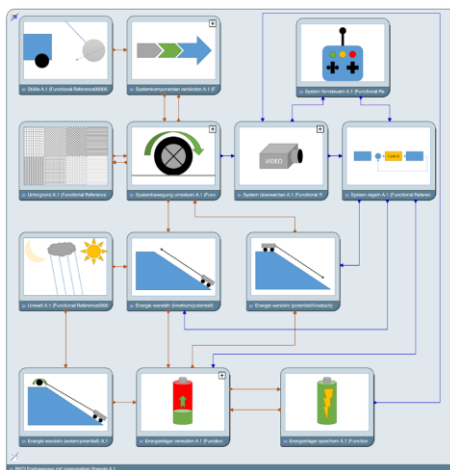


Bild 9. Funktionale Architektur des RECT in der 3DExperience Plattform [B116]



Die Darstellung der Konzeptfindung im morphologischen Kasten in der *3DExperience* Plattform ist wieder nur über nicht-modellbasierte Diagramme möglich.

Der Aufbau der Konzepte als logische Architekturen, die Verknüpfung mit funktionaler Architektur und die Integration der Simulationsmodelle gestaltet sich einfach. Es ist eine hohe Durchgängigkeit gegeben. Die Simulationsmodelle in der *3DExperience* Plattform sind logische Komponenten und damit Teile der logischen Architektur. Alle logischen Komponenten aus Bild 10 besitzen Simulationsmodelle. Ferner sind die logischen Komponenten mit den 3D-CAD-Modellen aus der *CATIA-V6*-App verknüpft und werden bei der Simulation berücksichtigt.

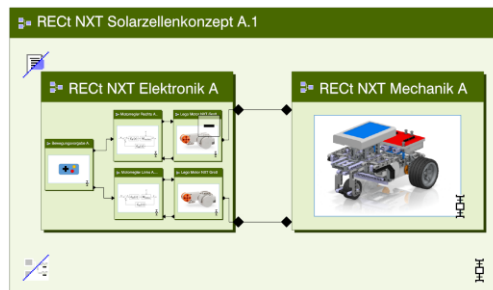


Bild 10. Logische Architektur und Simulationsmodell des RECT [B116]

Der Konzeptvergleich wird in der *3DExperience* Plattform mit der *Behavior Experience* App durchgeführt. Der Vergleich zwischen Anforderungen und Ergebnissen der Simulation kann in der verwendeten Version der *3DExperience* Plattform nur manuell durchgeführt werden. Trotzdem ist die Durchgängigkeit von Anforderung bis zum Ergebnis in der integrierten Lösung für das RECT vorhanden.

## 4 Ergebnisse

Integriertes und föderiertes MBSE sind mit heutigen Werkzeugketten möglich. Die Simulationsergebnisse sind vergleichbar. Der föderierte Ansatz ermöglicht den Einsatz der Werkzeuge, die für den jeweiligen Ansatz optimal sind. Er schafft auf der anderen Seite aber große Aufwände bei der Modellsynchronisation und erfordert vom Anwender ein hohes Maß an Disziplin bei der Modellerstellung und Modellverwaltung, inklusive einem technischen Verständnis für die eingesetzten Werkzeuge und deren Schnittstellen. Der integrierte Ansatz reduziert die Schnittstellenproblematik bei der Verwendung verschiedener Werkzeuge, hat aber den Nachteil, dass die Funktionalität der Werkzeuge teilweise nicht ausreicht. Dies schränkt heutige Methoden in ihrer Anwendung mit integrierten Werkzeugen ein. Unter der Voraussetzung, dass die Werkzeugfunktionalität und der Integrationsgrad in Zukunft weiter wachsen, wird dieser Nachteil verschwinden. Ferner muss überprüft werden, ob integrierte Werkzeuglösungen eine andere Vorgehensweise erfordern als föderierte Werkzeuglösungen. An den Stellen, an denen eine hohe Integration gegeben ist, sind integrierte Werkzeugumgebungen schon heute

wesentlich komfortabler, beispielsweise bei der Integration des 3D-CAD in die Simulation.

Für KARL MAYER würde die Verwendung einer integrierten Anwendung essentielle organisatorische Veränderungen bringen. Das Unternehmen hat heute die Gewissheit das MBSE auf integrierte und föderierte Weise möglich ist. Welchen Ansatz das Unternehmen weiterverfolgt, wird weiter untersucht und hängt von weiteren Faktoren ab.

Für die Hochschule Aschaffenburg hat sich die *3DExperience* Plattform als integriertes Werkzeug für die Lehre und für Forschungsprojekte bewährt. Integrierte Werkzeuge reduzieren die Aufwände bei der Schnittstellenpflege und ermöglichen mechatronische Systeme durchgängig zu modellieren und zu entwickeln, wie ein kürzlich durchgeführtes Forschungsprojekt zur Simulation und Validierung eines elektromechanischen Kettenschaltwerks zeigte [R117].

### Literaturverzeichnis

- [B116] Blumör, A.: Model Based Systems Engineering mit der 3DEXPERIENCE Plattform. Masterarbeit. Aschaffenburg: Hochschule Aschaffenburg 2016.
- [BPK15] Blumör A., Pregitzer, G., Kleiner, S., Neubert M. u. Hämisch, R.: Modellbasierte Entwicklung und vernetzter Produktentstehungsprozess im Maschinenbau. Tag des Systems Engineering. Ulm, 11.-13. November 2015. München: Hanser 2015, S. 153–162.
- [KI03] Kleiner, S.: Föderatives Informationsmodell zur Systemintegration für die Entwicklung mechatronischer Produkte. Forschungsberichte aus dem Fachgebiet Datenverarbeitung in der Konstruktion. Bd. 15. Aachen: Shaker 2013.
- [KK13] Kleiner, S. u. Kramer, C.: Model Based Design with Systems Engineering Based on RFLP Using V6. Bochum, Germany, March 11th-13th, 2013. Lecture notes in production engineering. Berlin, London: Springer 2013.
- [PBK14] Pregitzer, G., Blumör, A., Kleiner, S., Krastel, M. u. Neubert, M.: Model Based Systems Engineering: Einführung und Anwendung der modellbasierten Arbeitsweise in der Maschinenentwicklung. Tag des Systems Engineering. Bremen, 12. - 14. November 2014. München: Hanser 2015, S. 235–244.
- [VDI04] VDI 2206: Entwicklungsmethodik für mechatronische Systeme. Berlin: Beuth Verlag 2004.
- [Wt16] Weilkens, T. et al: Model-based system architecture. Wiley series in systems engineering and management. Hoboken, New Jersey: John Wiley & Sons, Inc 2016.
- [R117] Reinold, L.: Simulation und Validierung eines elektromechanischen Kettenschaltwerks mit Hilfe der 3DExperience Plattform. Masterarbeit. Aschaffenburg: Hochschule Aschaffenburg 2017.