

COMSim: Konzept zur Kompensation von Modelllücken in der simulationsgestützten Planungsphase von Produktionssystemen

E. Tinsel, A. Lechler und O. Riedel, ISW, Universität Stuttgart

1 Abstract

Die Realisierung der Virtuellen Inbetriebnahme bedingt die Erstellung vollständiger Simulationsmodelle. In der praktischen Umsetzung manifestieren sich jedoch regelmäßig Modelllücken. Die COMSim-Architektur, die in diesem Beitrag vorgestellt wird, adressiert dieses Problem durch ein modulares System, das fehlende Modellteile automatisiert erkennt und passende Ersatzlösungen vorschlägt. In diesem Kontext integrieren die Module "Wissenshub", "Datenaggregator" und "Assistenzsystem" logikbasierte und datengetriebene Verfahren, um aus unvollständigen Anforderungen simulationsfähige Modellvarianten zu generieren. COMSim ermöglicht somit frühzeitige Tests auch bei fragmentarischen Datenlagen und unterstützt eine adaptive, anforderungsbezogene Modellvervollständigung.

2 Einleitung

Der fortwährende digitale Wandel im industriellen Sektor fördert Innovation von und Umstrukturierung der konventionellen Planungsphase von Produktionssystemen. So formuliert der Verein Deutscher Ingenieure in seiner „VDI-Roadmap – Digitale Fabrik“ [1] insgesamt 7 Thesen, welche einen Ausblick auf eine mögliche Transformation der Planungsphase von Fabrikanlagen geben. Der Einsatz „neuer Planungsmodelle, -methoden und -werkzeuge“, ein stärkerer Fokus auf Anforderungsdaten aus der Auftragsabwicklung, ein Augenmerk auf KI-gestützte Assistenzsysteme und die virtuelle Fabrikplanung sind hierbei besonders hervorzuheben.

Die Vorzüge der Nutzung digitaler Methoden und Werkzeuge sind schon heute an industriellen Anwendungsbeispielen ersichtlich: So wird die Virtuelle Inbetriebnahme (VIBN) [2] als eine Methodik für das digitale Testen von Steuerungsverhalten genutzt. Grundlegend werden hierzu Simulationsmodelle aus den Ergebnissen der Planungsphase modelliert und innerhalb einer Simulationsumgebung an das entwickelte Steuerungsverhalten der Fabrikanlage angebunden. Aus welchen Bestandteilen sich ein Simulationsmodell für die Virtuelle Inbetriebnahme zusammensetzt kommt dabei ganz auf den Anwendungsfall und die definierten Testfälle an. So wird für eine Kollisionsprüfung immer auch ein Hüllkörper der beteiligten Anlagenelemente notwendig, für reine Ablauftests genügen dahingegen schon einfache Zustandsautomaten. Die Model- und Software-in-the-Loop-Testkonfigurationen der VIBN basieren dabei auf einer rein digitalen Nachbildung bzw. Emulation der Steuerung. Die Vorzüge eines solchen digitalen Tests sind evident – ohne Simulationsmodelle und virtueller Steuerung kann der Steuerungscode erst während der physischen Inbetriebnahme getestet werden. Fehler die in dieser späten Phase entdeckt werden, können schwerwiegend sein, wenn sie eine Rekonfiguration oder schlimmstenfalls Umplanung bedingen.

Da für die Durchführung der Virtuellen Inbetriebnahme ein Mehraufwand durch die Modellerstellung notwendig wird, empfehlen sich Ansätze der Wiederverwendbarkeit durch Modellbibliothekbildung [3, 4]. Weiter stellt die automatische Modellgenerierung (AMG) eine Methodik dar, um einen größtmöglichen Teil des Aufwands zu minimieren und einen beschleunigten Einstieg in die VIBN zu erlauben. Um ebendies zu ermöglichen, werden bereits vorhandene Modelle aus der Planungsphase mit Hilfe definierter Bausteine aus einer Modellbibliothek in VIBN-Modelle überführt.

In dieser Methodik verbirgt sich sehr viel Potenzial, denn die starke Kopplung von Planungsphase und simulativen Testen erlaubt es, früh unterschiedliche Prototypen zu evaluieren und gegen die festgelegten Anforderungen zu validieren und verifizieren. Ein möglichst hoher Automatisierungsgrad erlaubt dabei das iterative Testen verschiedener Lösungen und eröffnet die Möglichkeit, weitere KI-Ansätze, wie das Reinforcement Learning oder Generative KI, in diesen Prozessschritten einzusetzen. Die in **Abbildung 1** dargestellten Phasen sind wesentliche Bestandteile des Modellierungsprozesses bei der Erstellung von frühen Model-in-the-Loop-Simulationsmodellen. Die Modelle gründen sich dabei stets auf ein Planungsmodell (1), welches entweder unstrukturiert (natürliche Sprache, Bilder, Handzeichnungen) oder strukturiert (Datenformate, wie z.B. XML) vorliegen kann. Aus dem initialen Modell müssen nun die für die Simulation relevanten Simulationsmodelle extrahiert werden. Diese Modelle müssen für die jeweilige Simulationsumgebung entweder neu modelliert oder geeignete Komponenten aus einer Modellbibliothek bezogen und entsprechend parametrisiert werden (2). Besteht diese Bibliothek nicht aus einer Menge von Gesamtmodellen, sondern setzt sich die Simulationsumgebung aus Teilkomponenten zusammen, so sind diese im nächsten Schritt gegebenenfalls logisch zu verbinden (3). Es werden kinematische Ketten gebildet (z.B. Greifer und Roboter bewegen sich physikalisch im Verbund), Datenflüsse zwischen Sensoren und Steuerung abgebildet (z.B. eine Kamera triggert einen Sortiermechanismus), einfache Steuerungslogiken zugewiesen (z.B. Start/Stop-Verhalten oder Ablaufketten) oder logische Taktbeziehungen zwischen Komponenten definiert (z.B. Transportband darf erst anlaufen, wenn vorherige Bearbeitung abgeschlossen ist). Schließlich erfolgt ein Test des Gesamtmodells gegen die definierte Ablaufbeschreibung sowie eine Simulation von Prozesszuständen (z.B. Maschinenfehler, Wartezustände oder Blockaden bei Pufferung). Fehler, die sich aus dieser Simulation ergeben, müssen nun in einer Korrekturschleife behoben werden, bevor die korrigierte Version erneut getestet werden kann (5). Es ist evident, dass schnelle Iterationen nur möglich werden, wenn die Schritte 1-5 mit möglichst hohem Automatisierungsgrad durchgeführt werden können.

3 Stand der Technik

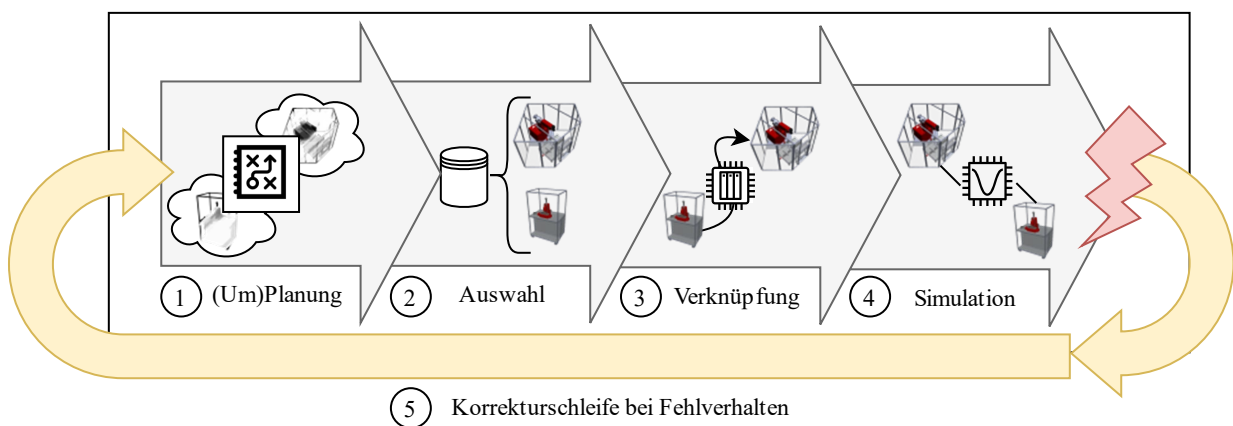


Abbildung 1: Schematische Darstellung einer Simulationsmodellerstellung

Die automatische Modellgenerierung im Bereich der Virtuellen Inbetriebnahme verwendet Daten aus der Planungsphase und überführt diese mittels regelbasierter Transformationen in ein VIBN-Modell. Müssen Teile des Modells noch manuell integriert werden, spricht man von einer semi-automatischen Modellgenerierung. Striffler und Voigt [5] haben in einem Sammelwerk unterschiedliche Ansätze der AMG verglichen. Unter Berücksichtigung dieser Konzepte [6–12] sind demnach folgende Prozessschritte für die AMG notwendig:

1. **Datenbasis bereitstellen:** Um eine Ableitung in ein VIBN-Modell zu ermöglichen, müssen für die AMG bereits strukturierte Formate aus vorangehenden Entwicklungsphasen vorliegen. Genannte Beispiele sind digitale Planungsdaten, wie Rohrleitungs- und Instrumentenfließschema oder Schaltpläne, Informationen aus Prozessleitsystemen, 3D-CAD-Modelle oder mittels Texterkennung digitalisierte Schriftstücke.
2. **Modelle mit Hilfe einer Modellbibliothek generieren und neue Modelle manuell integrieren:** Da die zugrundeliegenden Daten in strukturierter Form vorliegen, können diese maschinenlesbar verarbeitet und regelbasiert in VIBN-Modelle überführt werden (vgl. an einem Beispiel von Neyrinck et al. [13]). Für diese Generierung muss auf eine vorhandene Bibliothek von (Teil)Modellen der VIBN zurückgegriffen werden. Hierbei entscheidet der Informationsgehalt der zugrundeliegenden Daten über den Detailgrad des instanziierten VIBN-Modells – fehlen Informationen zu notwendigen Parametern oder Varianten eines Modells in der zugrundeliegenden Datenbasis, so muss die Auswahl oder Parametrierung auf Grundlage von Standardbelegungen erfolgen und gegebenenfalls nachträglich vom Benutzer angepasst werden; falls eine Abbildung der Datenbasis in die VIBN-Modellbibliothek auf Komponentenebene nicht vollständig ist; so muss das Modell der Bibliothek manuell hinzugefügt werden und mit der Datenbasis über neue Transformationsregeln verbunden werden. Dies gilt auch für eine unvollständige Datenbasis. Beinhaltet diese beispielsweise Informationen über die einzelnen Entitäten, nicht jedoch deren Komposition, so muss diese Verknüpfung entweder manuell hinzugefügt werden, oder es existiert bereits eine logisch ableitbare Verknüpfungsregel in einem bestehenden Regelsatz.
3. **Simulationsexperiment durchführen:** Die Simulation wird schließlich in einer visuell dargestellten Simulationsumgebung ausgeführt und die Testergebnisse aufgezeichnet. Auf Basis dieser Daten können Rückschlüsse auf das Verhalten einzelner Komponenten sowie das Zusammenwirken verknüpfter Steuerungseinheiten gezogen werden. Wird ein fehlerhaftes Verhalten identifiziert, ist eine Anpassung des Simulationsmodells erforderlich, woraufhin ein weiteres Simulationsexperiment durchgeführt wird. Unter Einsatz der AMG werden die Änderungen auf der Datenbasis manuell durchgeführt und in ein neues VIBN-Modell überführt.

4 Forschungsbedarf und Ziele

Die grundlegende Anforderung an eine wirtschaftliche VIBN ist deren effiziente Erstellung. Schon jetzt hilft die AMG bei einer Beschleunigung und fördert die Wiederverwendbarkeit von Komponenten.

Eine potenzielle Schwachstelle manifestiert sich insbesondere in Situationen, in denen die Datengrundlage nicht vollständig mit Hilfe der VIBN-Modellbibliothek abgebildet werden können. In derartigen Fällen ist grundsätzlich manueller Entwicklungsaufwand erforderlich. Das Problem zeigt sich in sämtlichen Vorhaben, die von der Serienfertigung weg, hin zum Sondermaschinenbau tendieren und folglich einen Zeitaufwand für die Neumodellierung oder Anpassung von Modellen bedingen.

Dabei bedeutet eine unvollständige VIBN-Modellbibliothek nicht zwangsläufig, dass die Modelle nicht bereits modelliert wurden: Arbeiten mehrere Maschinenbauer an einem gemeinsamen Vorhaben, so ist es möglich, dass ein Austausch der firmeninternen Modelle nicht möglich ist; ein Test der gesamten Anlage oder Zelle wird somit erschwert. Gründe hierfür könnten Datenschutzbedenken oder ein zu hoher Aufwand für die Konvertierung zwischen unterschiedlichen Simulationswerkzeugen der Unternehmen sein. Sind einzelne Modelle im Gesamtmodell generell noch nicht modelliert, so besteht dieses Problem auch in einem einzelnen Unternehmen mit mehreren Simulationsmodellentwicklern, die ihre Teillösungen nicht im Gesamtmodell validieren können, bis

dieses vollständig vorliegt. Das Problem bleibt bestehen, wenn bei dem Gesamtablauf Probleme auftreten, da nun wieder einzelne Modelle angepasst werden, die den nächsten Test des Gesamtablaufs verzögern (vgl. **Abbildung 2**).

Dieser Aspekt ist ein Faktor, der eine Öffnung hin zu KI-getriebenen Ansätzen motiviert. Der Personal- und Zeitaufwand erscheint insbesondere für mittelständische Unternehmen als nicht wirtschaftlich.

In diesem Zusammenhang ist es von entscheidender Bedeutung, die Automatisierbarkeit der Prozesse zu erforschen und die automatische Modellgenerierung weiter zu verbessern. Dabei ist zu berücksichtigen, dass der Entwicklungsaufwand nicht nur in die Detaillierung vorangehender Daten verschoben wird. Ein signifikanter Zeitgewinn kann nur erzielt werden, wenn Methoden und Technologien die Modellierung projektübergreifend unterstützen.

Aus den genannten Gründen ist eine Anwendung von AMG nur unter der Voraussetzung einer vollständigen Verfügbarkeit von strukturierten Basisdaten möglich. Im Folgenden wird ein Konzept vorgestellt, das einen Test des Gesamtablaufs eines Simulationsprojekts bereits dann ermöglicht, wenn einzelne Simulationskomponenten zwar in einer unstrukturierten Prozessbeschreibung definiert sind, in der Simulation als konkrete Instanz jedoch fehlen. Um dies zu ermöglichen, müssen drei Teilziele adressiert werden.

1. **Erkennung zusammenhängender Simulationsmodelle im Gesamtmodell**

Um fehlende Teilkomponenten eines Gesamtmodells zu erkennen, muss ein Ziel die automatische Erkennung von Zusammenhängen zwischen den einzelnen Modellen sein. Die AMG kann zwar Zusammenhänge 1-zu-1 abbilden, falls die Datenbasis diese bereits vollständig (also mit allen für die VIBN notwendigen Simulationsaspekten, wie z.B. Verhalten oder CAD-Beschreibung) für die Transformation vorgibt, das Ziel dieser Methodik soll jedoch sein, auch für un spezifizierte Vorhabensbeschreibungen eine Lösung zu finden, da gerade diese ja implizit eine Kompensation von Modelllücken voraussetzen.

2. **Abbilden des Verhaltens erkannter Simulationsmodelle im Gesamtmodell**

Ist das Ziel der Identifikation von Modelllücken erfüllt, müssen diese durch einen Kompensationsmechanismus geschlossen werden. Die Voraussetzung ist dabei eine möglichst genaue Nachbildung des vorgesehenen Modells abzuleiten. Das Ziel gilt als erfüllt, falls ein Ersatzmodell gefunden werden kann, welches in seinem Verhalten und seiner Struktur dem intendierten Modell ähnelt. Hierbei soll in einem ersten Schritt nicht bemessen werden, wie exakt ein Ersatzmodell sein Original abbildet, sondern eine Methodik gefunden werden, welche aus einer Menge an Modellen in einer Modellbibliothek das geeignetste Modell auswählt.

3. **Früheres Testen von Gesamtabläufen**

Schließlich besteht ein übergeordnetes Ziel darin, dass auch unter Einsatz der Modellkompensation ein Testen des Gesamtablaufs der Simulation möglich ist. Im Vordergrund steht dabei ein hoher Grad an Automatisierung, sodass die Methode keinen Nachteil gegenüber dem Stand der Technik unter Verwendung der AMG hat. Da der Lösungsraum für ein Ersatzmodell an einer identifizierten Modelllücke nicht zwangsläufig nur eine eindeutige Lösung beinhaltet, muss die Methodik dem Entwickler zudem die Möglichkeit einer Variantenbildung ermöglichen. Es ist essenziell, dass trotz des Ersatzmodells ein Testergebnis aller Simulationskomponenten und des Gesamtablaufs durch das Simulationsexperiment erstellbar bleibt. Der manuelle Aufwand für die Integration der Ersatzmodelle in das Gesamtmodell soll stets minimal gehalten werden.

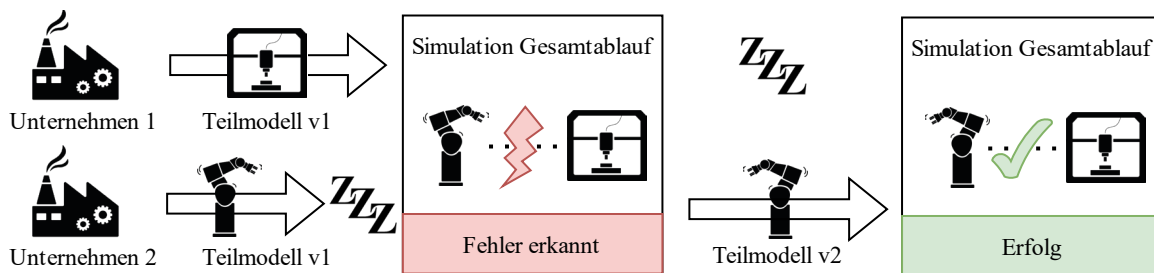


Abbildung 2: Das Testen an einem Gesamtmodell verzögert sich, solange nicht alle Teilkomponenten zur Verfügung stehen

Zur Veranschaulichung der zuvor genannten Ziele wird im Folgenden auf ein Beispiel verwiesen (vgl. **Abbildung 3**): Im Rahmen der Vorbereitung der Inbetriebnahme ist die Durchführung eines simulativen Tests einer **Vereinzelungsstation** des **Herstellers A** in Kombination mit einem **Transportband** des **Herstellers B** vorgesehen. **Hersteller A** verfügt bereits über ein Simulationsmodell seiner Station, während von **Hersteller B** lediglich eine Produktspezifikation vorliegt. Um eine valide Testprozedur für den Gesamtablauf und die Übergabemechanismen zwischen den beiden Simulationsmodellen zu gewährleisten, wäre es erforderlich, dass der **Hersteller B** das **Transportband** aufwändig nachmodelliert.

Um eine automatisierte Lösung für dieses Problem zu entwickeln, ist zunächst die Schaffung eines Mechanismus erforderlich, der erkennt, dass die **Vereinzelungsstation** von **Hersteller A** 1) ein **Transportband** voraussetzt und 2) wie das **Transportband** logisch mit der **Vereinzelungsstation** verbunden ist (vgl. Ziel 1). Im Anschluss wird eine Methodik benötigt, die auf Basis der Produktspezifikation von **Hersteller B** ein Ersatzmodell ableitet, das das Verhalten des realen **Transportbands** möglichst genau abbildet. Es besteht die Möglichkeit, dies beispielsweise durch die Selektion eines vergleichbaren Modells aus der Bibliothek des **Herstellers A** oder durch ein generisches Kompensationsmodell in Form eines quaderförmigen Ersatzverhaltens zu realisieren. Letzteres bewegt den Werkstückträger entsprechend der spezifizierten Geschwindigkeit horizontal unter die Station (vgl. Ziel 2).

Im finalen Schritt ist eine automatische Anbindung des Ersatzmodells an das Verhalten der **Vereinzelungsstation** zu gewährleisten. Um ein optimales Ergebnis zu erzielen, ist es notwendig, den Zeitpunkt der Vereinzelung mit der Position des Materialstückträgers auf dem Transportband abzustimmen. Zudem ist es erforderlich, den Materialstückträger unter der Station so lange zu stoppen, bis die Vereinzelung abgeschlossen ist (vgl. Ziel 3).

5 Konzept

Um die drei aufgestellten Ziele zu erreichen, wird im Folgenden das Konzept der COMSim-Architektur vorgestellt, die ein Anwendungsbeispiel für die Methodik der Modellkompensation darstellt. COMSim unterteilt sich grundlegend in vier Module: **Wissenshub**, **Datenaggregator**, **Assistenzsystem** und **Simulationsumgebung**.

Das Modul **Wissenshub** stellt eine Datenbasis als Quelle für die Entscheidungslogik bereit, welche als Eingabedaten aufbereitete Beschreibungsdaten eines bestehenden Simulationsmodells verarbeitet und einen Lösungsraum bestehend aus integrierbaren Teilmodellen zurückliefert. Die Wirkweise dieses Moduls wird für das Konzept im Detail dargelegt.

Die Vorbereitung der Wissensbasis mit möglichst umfangreichen Daten erfordert die Implementierung eines **Datenaggregators**, dessen Funktion darin besteht, Informationen aus internen Datenquellen gezielt für die Wissensbasis vorzuverarbeiten und zu integrieren. Darüber

hinaus ist eine Untersuchung externer Datenquellen, wie etwa Standards oder offene Informationsquellen, vorgesehen, deren Ergebnisse ebenfalls in die Wissensbasis integriert werden sollen.

Ein der Simulationsumgebung übergeordnetes Modul Assistenzsystem den Bediener bei der Selektion geeigneter Teilmodelle aus dem Lösungsraum, hilft bei der Einschränkung von diesem und erlaubt für mehrdeutige Lösungen die Darstellung von Modellvarianten. Das Ergebnis wird schließlich über eine AMG in die Simulationsumgebung zurücktransformiert. Abbildung 4 zeigt die drei Hauptmodule sowie deren Komponenten, die in den folgenden Abschnitten einer detaillierten Betrachtung unterzogen werden.

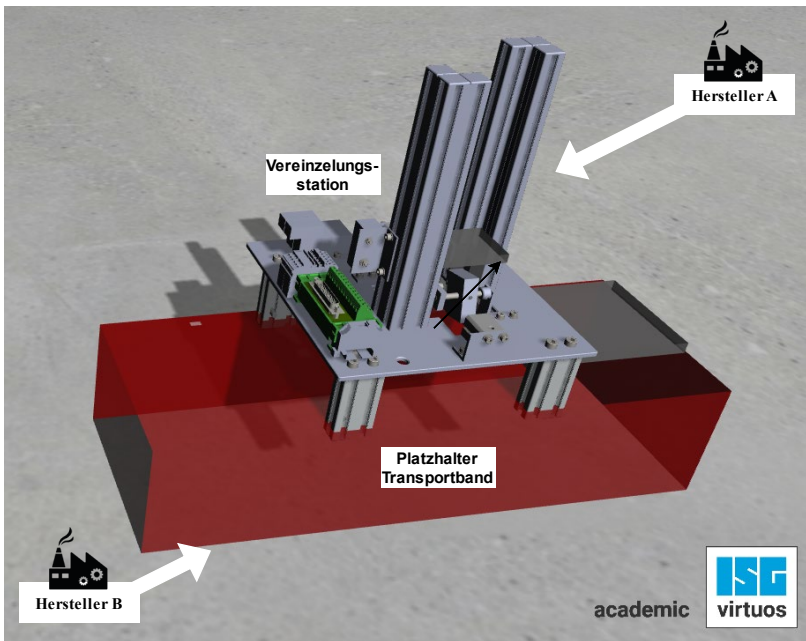


Abbildung 3: Beispiel einer Lucas-Nülle Vereinzelungsstation (Hersteller A) mit fehlendem Transportband (Hersteller B), welches durch einen Platzhalter ersetzt wurde

Angepasste Simulationsumgebung mit Import- und Exportschnittstelle

Es wird angenommen, dass die Simulationsumgebung bereits Simulationsexperimente durchführen kann und diese auf Grundlage von zusammengesetzten Simulationsmodellen operieren. Hierdurch wird für wiederkehrende Aufgaben der Bedarf einer Modellbibliothek evident. Aus dieser sollen die jeweiligen Teilkomponenten bezogen werden können. Um eine Zuordnung zu ermöglichen muss diese Modellbibliothek über eine Schnittstelle nach Modellen durchsuchbar sein. Die Suche soll auf Basis von Metadaten durchführbar sein, welche den Simulationsmodellen anhängig sind. Die Modellbibliothek beinhaltet konkrete Instanzen von Simulationsmodellen mit konkreter Ausprägung von Verhalten, CAD und Schnittstellen. Um das Konzept der Kompensation zu ermöglichen, ist eine zusätzliche Bibliothek für rudimentäres Ersatzverhalten, wie Physik- und/oder Kinematikverhalten, notwendig. Diese Minimalbausteine sollen für das Konzept genutzt werden, um ein Verhalten nachzubilden, falls kein konkretes Modell in der Modellbibliothek vorliegt, welches als Ersatzmodell geeignet ist.

Die Simulationsumgebung muss zudem über Schnittstellen für einen automatischen Import und Export verfügen. Dabei sind die Parametrisierung, das Layout und die Verknüpfung von Teilkomponenten für die vor- und nachgelagerte Verarbeitung von besonderer Bedeutung.

Um die Ergebnisse aus den Simulationsexperimenten in die Variantenbeurteilung des Assistenzsystems einfließen zu lassen, muss die Simulationsumgebung zudem über Schnittstellen zum automatischen Instanzieren und Durchführen der Simulation erlauben.

Schließlich soll die Simulationsmodellerstellung des Entwicklers über ein Beobachterpattern aufgezeichnet werden, sodass erstellte Modelle, deren Verknüpfung und Parametrierung gegeben den Randbedingungen übriger Modelle in das Wissensnetz einfließen können, was in ein selbstlernendes System resultiert. Die Schnittstellen des Moduls werden von dem Modul Wissenshub und Assistenzsystem genutzt. Modelle werden dabei vom Datenextraktor exportiert und der Modelltransformator des Assistenzsystems erstellt ein geeignetes Importformat für die Importschnittstelle der Simulationsumgebung.

Wissenshub zur anforderungsbezogenen Lösungsfindung

Der Wissenshub stellt die datenverarbeitende und entscheidungsunterstützende Komponente des Gesamtsystems dar. Er nimmt die Funktion einer Schnittstelle ein, welche die exportierten Modelle der Simulationsumgebung, die domänenspezifische Wissensbasis und das Assistenzsystem zur Variantenbewertung und Modellauswahl miteinander verbindet. Zunächst überführt ein Datenextraktor die aus der Simulationsumgebung exportierten Modelle. Die Modelle beinhalten sowohl strukturelle Informationen (Komponenten, Schnittstellen, Verbindungen) als auch semantische Inhalte (Parameter, Modi, Verhaltenserwartungen). Der Extraktor bereitet diese Informationen so auf, dass sie in eine formalisierte Repräsentation überführt werden können, etwa als Graphstruktur oder Feature-Vektor.

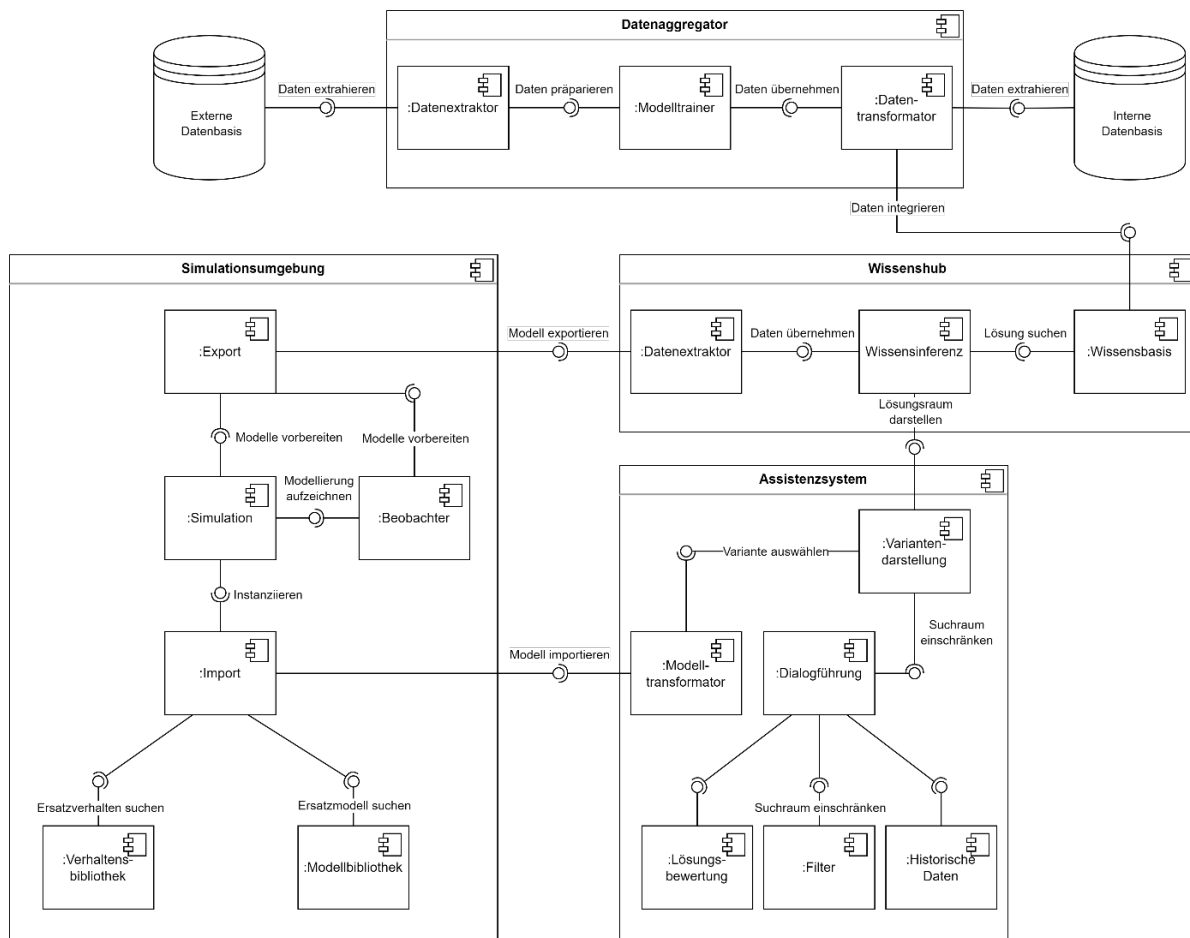


Abbildung 4: Architektur COMSim

Subsektiv wird die Analyse der extrahierten Daten von der Wissensinferenz vorgenommen und in eine Suche innerhalb der Wissensbasis umgewandelt. Im Rahmen dieses Prozesses werden die extrahierten Modellkomponenten mit bereits vorhandenen Lösungsmustern abgeglichen, um funktional oder strukturell passende Ergänzungen zu identifizieren. Die Wissensinferenz nutzt einerseits logikbasierte Schlussfolgerungsverfahren, beispielsweise regelbasierte Inferenzsysteme auf Basis von OWL-Ontologien oder First-Order-Logic, um explizit definierte Beziehungen zwischen Komponenten, Schnittstellen und Verhaltensmustern zu analysieren. Andererseits kommen strukturorientierte Vorschlagsmechanismen auf Basis von Graph-Neural-Networks (GNNs) zum Einsatz, die auch ohne klassische gelabelte Trainingsdaten operieren können. Statt überwacht zu lernen, werden GNNs in einem self-supervised oder unsupervised Setting verwendet, um semantisch bedeutsame Embeddings von Teilgraphen zu erzeugen. Diese Embeddings dienen anschließend der Ähnlichkeitsbewertung zwischen einer Modelllücke und bekannten Strukturen in der Wissensbasis. Ergänzend können wenige manuell kuratierte Beispielmuster für ein few-shot-artiges Training genutzt werden, oder der GNN wird im laufenden Systembetrieb durch Nutzerfeedback iterativ verbessert. Dadurch ist die Wissensinferenz in der Lage, auch bei unvollständigen oder unspezifisch formulierten Anforderungen funktional passende Lösungskandidaten vorzuschlagen, die strukturell plausibel und inhaltlich kompatibel mit dem bestehenden Simulationsmodell sind. Die Kombination aus regelbasierter Logik und graphbasierter Strukturähnlichkeit ermöglicht dabei eine robuste und anpassungsfähige Erschließung des Lösungsraums. Diese Methoden ermöglichen es, auch dann geeignete Modellkomponenten oder Konfigurationsmuster vorzuschlagen, wenn die Anforderungsbeschreibung unvollständig, mehrdeutig oder nur implizit vorhanden ist.

Der Wissenshub erhält strukturierte Exportdaten (z. B. XML, JSON oder spezifische DSL-Formate) direkt aus der Simulationsumgebung über die Schnittstelle „Modell exportieren“. Diese Daten bilden die Grundlage für die nachgelagerte Analyse und Lösungsfindung. Die Verbindung zur internen und externen Datenbasis erfolgt über gemeinsame Datenformate und Transformationsprozesse. Der Datenextraktor im Wissenshub kann insbesondere auf vorverarbeitete Daten des Datenaggregators zurückgreifen.

Es besteht beispielsweise die Möglichkeit, bereits angereicherte Komponenteninformationen, ML-Modelle oder Metadaten aus dem Modelltrainer oder Datentransformator zu integrieren, um die Lösungsfindung zu verfeinern. Die Resultate der Wissensinferenz, also potenzielle Lösungskomponenten oder Varianten, werden in strukturierter Form an das Assistenzsystem übergeben. Die Übergabe erfolgt an die Variantendarstellung, welche die Vorschläge visualisiert und interaktiv auswählbar macht. Gleichzeitig kann das Assistenzsystem die Rückkopplung aus der Dialogführung, den Filtern oder der Lösungsbewertung nutzen, um die Suchparameter der Wissensinferenz iterativ einzugrenzen. In der Konsequenz manifestiert sich eine bidirektionale Schnittstelle, in welcher Nutzerentscheidungen unmittelbar Einfluss auf die Wissensabfrage und -verarbeitung nehmen können.

Architektur zur Aggregation heterogener Datenquellen

Der Datenaggregator fungiert als zentrales Bindeglied zwischen externen und internen Datenquellen sowie der Wissensbasis des Systems. Seine Hauptaufgabe besteht in der Aufbereitung domänenrelevanter Informationen sowie deren Bereitstellung in strukturierter Form für den Wissenspool. Das Vorgehen ist durch einen mehrstufigen Verarbeitungsprozess charakterisiert, der auf drei Kernkomponenten basiert.

Der Datenextraktor ist zuständig für die initiale Gewinnung von Rohinformationen aus externen Datenquellen (z. B. technische Standards, Fachliteratur, öffentlich zugängliche Datenbanken) sowie aus internen Datenquellen (z. B. interne Systemhistorien, Projektdokumentationen). Der vorliegende Datensatz beinhaltet sowohl strukturierte als auch unstrukturierte Inhalte, zu denen beispielsweise

technische Parameter, Konfigurationsbeispiele, Erfahrungsberichte oder semantische Beschreibungen zählen.

Die extrahierten Rohdaten werden anschließend einem Modelltrainer übergeben, der diese mit Hilfe lernbasierter Verfahren – insbesondere durch Einsatz großer Sprachmodelle (LLMs) – analysiert (vgl. Tinsel et al. [14]). Das Ziel besteht darin, Korrelationen, Abhängigkeiten, Funktionsmuster und semantische Beziehungen zwischen Komponenten oder Systemverhalten zu identifizieren. Der Modelltrainer generiert daraus abstrahierte Wissensbausteine, die sich zur Wiederverwendung in modellbasierten Lösungsräumen eignen. Als Beispiele können typische Anschlussbeziehungen, Konfigurationsmuster oder standardisierte Funktionsmodule angeführt werden.

Der Datentransformator übernimmt die Aufgabe, die vom Modelltrainer erzeugten oder angereicherten Inhalte in ein einheitliches, strukturiertes Format zu überführen. Im Rahmen dieses Prozesses erfolgt eine Transformation der Informationen in eine Form, die eine Kompatibilität mit der Ontologie und den Datenstrukturen der Wissensbasis gewährleistet. Dubletten, Inkonsistenzen und Mehrdeutigkeiten werden eliminiert, sodass eine automatische und verlustfreie Integration der Daten gewährleistet ist.

Der Datenaggregator weist zwei separate Eingangsströme für Datenquellen auf. Diese fungieren als Rohmaterial für die Extraktion und nachfolgende Transformation. Der Aggregator ermöglicht somit sowohl die Nutzung internen Know-hows als auch die Erschließung aktueller Erkenntnisse aus externen, offenen Quellen.

Am Ende der Aggregationskette wird der Transfer der bereinigten und formalisierten Wissensseinheiten in die Wissensbasis vollzogen. Die Integration erfolgt über eine gemeinsame Datenrepräsentation, wodurch sichergestellt wird, dass das im System verfügbare Wissen stets aktuell, erweiterbar und maschinenlesbar vorliegt.

Funktion eines adaptiven Assistenzsystems

Das Assistenzsystem bildet die interaktive Schnittstelle zwischen den vom Wissenshub bereitgestellten Lösungsvorschlägen und der letztlich vom Benutzer akzeptierten oder modifizierten Modellvariante. Ziel ist es, aus einem potenziell großen Lösungsraum gezielt jene Varianten herauszufiltern und zu bewerten, die sich für die Vervollständigung eines unvollständigen oder fehlerhaften Simulationsmodells besonders gut eignen. Dabei wird berücksichtigt, dass für eine gegebene Modelllücke in der Regel mehrere funktional mögliche, aber unterschiedlich geeignete Alternativen existieren.

Zunächst erfolgt eine strukturierte Aufbereitung und Visualisierung des ermittelten Lösungsraums über die Komponente der Variantendarstellung. Diese zeigt dem Benutzer verschiedene Alternativen zur Modellergänzung, etwa unterschiedliche Komponenten, Parametrisierungen oder strukturelle Integrationen. Die Auswahl dieser Varianten erfolgt nicht automatisiert, sondern im engen Zusammenspiel mit der Komponente zur Dialogführung. Diese führt den Benutzer durch den Entscheidungsprozess, stellt gezielte Rückfragen zur technischen Zielsetzung oder den Systemanforderungen und ermöglicht es, den Suchraum iterativ einzuschränken.

Die individuelle Nutzereingabe sowie weitere kontextuelle Informationen – etwa bestehende Systemkonfigurationen oder Rahmenbedingungen – werden genutzt, um Filtermechanismen anzustoßen, die den Lösungsraum weiter zu verkleinern. Parallel dazu greift das System auf eine Bewertungskomponente zurück, die auf Grundlage definierter Metriken wie Kompatibilität, Komplexität, Reifegrad oder Wiederverwendbarkeit eine Priorisierung der Varianten vornimmt. Die Bewertung kann sowohl regelbasiert als auch datengetrieben erfolgen, etwa gestützt auf Performancekennzahlen aus früheren Modellsimulationen.

Ein zentraler Aspekt des Systems ist die Fähigkeit zur adaptiven Entscheidungsunterstützung. Dies manifestiert sich insbesondere in der Einbindung historischer Daten, die in Form früherer Nutzerentscheidungen vorliegen. Diese Informationen werden in die Vorschlagspriorisierung einbezogen, was die Wahrscheinlichkeit erhöht, eine individuell geeignete Variante vorzuschlagen. Nach erfolgter Entscheidung des Benutzers für eine Variante folgt die Überführung des gewählten Lösungselements in das bestehende Simulationsmodell mittels der Komponente des Modelltransformators. Im Zuge dessen werden sowohl strukturelle als auch semantische Integrationen vorgenommen, sodass ein konsistentes, simulierbares Modell entsteht.

Zusammenfassung & Ausblick

Das vorgestellte Konzept COMSim liefert ein fundiertes System zur Kompensation von Modelllücken in der simulationsgestützten Planungsphase. Der zentrale Gedanke besteht in der Fähigkeit, auch bei unvollständiger oder unspezifischer Ausgangslage simulationsfähige Modelle zu erzeugen, um dadurch frühzeitig das Systemverhalten validieren zu können. Der Fokus liegt auf dem modularen Aufbau, der den Wissenshub zur strukturierten Analyse und Suche einbettet, den Datenaggregator zur kontinuierlichen Anreicherung der Wissensbasis nutzt und ein Assistenzsystem bereitstellt, das den Benutzer aktiv in die Lösungsfindung einbindet. Die angewandten Mechanismen umfassen regelbasierte Inferenz, graphbasierte Strukturähnlichkeitsanalysen und die Nutzung von LLMs zur semantischen Interpretation von Rohdaten.

Es sei darauf hingewiesen, dass COMSim nicht nur auf vorhandene Bibliothekseinträge zugreift, sondern aktiv Variantenbildung betreibt und aus einer Vielzahl potenzieller Kompensationslösungen jene auswählt, die dem gegebenen Kontext am besten entsprechen. Die Einbindung historischer Nutzungsdaten, interaktiver Filtermechanismen sowie automatisierter Modelltransformationen erlaubt eine flexible und adaptiv lernende Systemlogik, die sich auch in heterogenen oder wenig standardisierten Produktionsumgebungen anwenden lässt.

Für den weiteren Forschungs- und Entwicklungsverlauf ergeben sich mehrere Ansatzpunkte: Zunächst ist eine Verfeinerung der Bewertung von Ersatzmodellen erforderlich, um sowohl strukturelle als auch verhaltensbasierte Kompatibilitäten präziser abzubilden. Darüber hinaus bietet die automatische Parametrierung und Verhaltenserzeugung in Verbindung mit Reinforcement-Learning-Strategien, bei denen Ersatzmodelle durch Simulation iterativ optimiert werden können, ein erhebliches Potenzial. Darüber hinaus erscheint die bidirektionale Rückkopplung zwischen realer und virtueller Inbetriebnahme vielversprechend. So könnten etwa Ergebnisse aus dem realen Anlagenbetrieb genutzt werden, um die Wissensbasis kontinuierlich zu verbessern, etwa durch Feedback-Loops oder automatisierte Modellkorrekturen.

Langfristig kann die hier dargestellte Architektur den Weg zu vollständig KI-gestützten Modellierungs- und Testverfahren ebnen, die nicht nur auf Daten reagieren, sondern auch in der Lage sind, eigenständig Vorschläge zu entwickeln, diese zu bewerten und mittels Simulation zu validieren. Insbesondere für komplexe oder wandlungsfähige Produktionssysteme mit hoher Variantenvielfalt bietet COMSim somit eine Perspektive für robuste, skalierbare und adaptive Planungsprozesse.

References

- [1] VDI-Gesellschaft Produktion und Logistik (VDI-GPL), Die digitale Fabrik - Treiber der digitalen Transformation, 2023. Accessed: Jun. 12 2025. [Online]. Available: <https://www.vdi.de/ueber-uns/presse/publikationen/details/die-digitale-fabrik-treiber-der-digitalen-transformtion>.

- [2] G. Wünsch, *Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme*. München: Herbert Utz Verlag, 2008. Accessed: Jun. 12 2025. [Online]. Available: <https://www.utzverlag.de/assets/pdf/407951es.pdf>.
- [3] M. Hussain, N. Masoudi, G. Mocko, and C. Paredis, "Approaches for Simulation Model Reuse in Systems Design — A Review," *SAE Int. J. Adv. & Curr. Prac. in Mobility*, vol. 4, no. 5, pp. 1457–1471, 2022, doi: 10.4271/2022-01-0355.
- [4] Y. Liu, L. Zhang, W. Zhang, and X. Hu, "An overview of simulation-oriented model reuse," in *Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems: 16th Asia Simulation Conference and SCS Autumn Simulation Multi-Conference, AsiaSim/SCS AutumnSim 2016, Beijing, China, October 8-11, 2016, Proceedings, Part IV 16*, pp. 48–56.
- [5] N. Striffler and T. Voigt, "Concepts and trends of virtual commissioning – A comprehensive review," *Journal of Manufacturing Systems*, vol. 71, pp. 664–680, 2023, doi: 10.1016/j.jmsy.2023.10.013.
- [6] *Automatisch generierte Simulationsmodelle verfahrenstechnischer Anlagen für den Steuerungstest*, 2011. [Online]. Available: https://www.researchgate.net/profile/mike-barth/publication/236656161_automatisch_generierte_simulationsmodelle_verfahrenstechnischer_anlagen_fur_den_steuerungstest/links/0deec518f5b05df766000000/automatisch-generierte-simulationsmodelle-verfahrenstechnischer-anlagen-fuer-den-steuerungstest.pdf.
- [7] E. Arroyo, M. Hoernicke, P. Rodríguez, and A. Fay, "Automatic derivation of qualitative plant simulation models from legacy piping and instrumentation diagrams," *Computers & Chemical Engineering*, vol. 92, pp. 112–132, 2016, doi: 10.1016/j.compchemeng.2016.04.040.
- [8] M. Barth and A. Fay, "Automated generation of simulation models for control code tests," *Control Engineering Practice*, vol. 21, no. 2, pp. 218–230, 2013, doi: 10.1016/j.conengprac.2012.09.022.
- [9] M. Barth, M. Strube, A. Fay, P. Weber, and J. Greifeneder, "Object-oriented engineering data exchange as a base for automatic generation of simulation models," in *2009 35th Annual Conference of IEEE Industrial Electronics*, 2009, pp. 2465–2470.
- [10] O. Mathias, W. Gerrit, D. Oliver, L. Benjamin, S. Markus, and U. Leon, "Automatic Model Generation for Virtual Commissioning based on Plant Engineering Data," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11635–11640, 2014, doi: 10.3182/20140824-6-za-1003.01512.
- [11] G. Santillan Martinez, S. A. Sierla, T. A. Karhela, J. Lappalainen, and V. Vyatkin, "Automatic Generation of a High-Fidelity Dynamic Thermal-Hydraulic Process Simulation Model From a 3D Plant Model," *IEEE Access*, vol. 6, pp. 45217–45232, 2018, doi: 10.1109/access.2018.2865206.
- [12] S. Sierla, M. Azangoo, A. Fay, V. Vyatkin, and N. Papakonstantinou, "Integrating 2D and 3D Digital Plant Information Towards Automatic Generation of Digital Twins," in *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, 2020, pp. 460–467.
- [13] A. Neyrinck, A. Lechler, and A. Verl, "Automatic Variant Configuration and Generation of Simulation Models for Comparison of Plant and Machinery Variants," *Procedia CIRP*, vol. 29, pp. 62–67, 2015, doi: 10.1016/j.procir.2015.02.069.

- [14] Concept of an Initial Requirements-Driven Factory Layout Planning and Synthetic Expert Verification for Industrial Simulation Based on LLM: IEEE, 2024.

©  CC BY 4.0

<https://creativecommons.org/licenses/by/4.0/deed.de>