

Systematische Entwicklung einer Architektur für die virtuelle Inbetriebnahme eines automatisierten Produktionssystems

Omar Ismail, Lasse Beers, Felix Gehlhoff

Institut für Automatisierungstechnik

Helmut-Schmidt-Universität/Universität der Bundeswehr Hamburg

Hamburg, Deutschland

omar.ismail@hsu-hh.de

Nihar H. Shah

Airbus Operations GmbH

Hamburg, Deutschland

nihar.shah@airbus.com

Alexander Fay

Lehrstuhl für Automatisierungstechnik

Ruhr-Universität Bochum

Bochum, Deutschland

alexander.fay@rub.de

Zusammenfassung—Die Unternehmen der Fertigungsindustrie stehen heute unter erheblichem Druck, einerseits hohe Qualität bei komplexen Produktionsprozessen sicherzustellen und andererseits einen schnellen Ratenhochlauf zu gewährleisten. Um diese Herausforderungen effizient und fehlerfrei zu bewältigen, gewinnt die Virtuelle Inbetriebnahme (VIBN) immer mehr an Bedeutung. Eine VIBN-Architektur ermöglicht eine tiefgehende Analyse des SPS-Codes und des Ressourcenverhaltens mithilfe von 3D-Modellen und Steuerungsimulationen vor der tatsächlichen Implementierung, wodurch Fehler während der Inbetriebnahme signifikant reduziert werden können. Das Fehlen von standardisierten Vorgehensweisen für die Entwicklung solcher VIBN-Architekturen stellt dennoch eine Herausforderung dar. Ziel dieses Beitrags ist die Entwicklung einer anpassungsfähigen VIBN-Architektur, die speziell auf die Anforderungen der Flugzeugfertigung zugeschnitten ist. Dabei zeigt der Beitrag die Wirksamkeit der Anwendung der SPES-Methodik (Software Platform Embedded Systems) zur Entwicklung einer solchen Architektur auf und identifiziert gleichzeitig Bereiche zur Verbesserung, insbesondere bei der Entscheidungsunterstützung. Der Beitrag bietet damit einen Leitfaden für Forscher und Praktiker im Bereich der VIBN von automatisierten Produktionssystemen, indem er eine strukturierte Vorgehensweise zur Entwicklung komplexer Systeme aufzeigt.

Stichworte—Virtuelle Inbetriebnahme, Methodik der Software Platform Embedded Systems, modellbasierte Systementwicklung, automatisierte Produktionssysteme, Flugzeugfertigung

I. EINLEITUNG

Die Fertigungsindustrie und insbesondere die Flugzeugfertigung zeichnen sich durch ein hohes Maß an Präzision bei gleichzeitiger Forderung nach hoher Qualität, Effizienz und Anpassungsfähigkeit bei der Montage komplexer Bauteile aus. Gleichzeitig besteht bei der Einführung neuer Produktionssysteme ein hoher Druck, die geforderten Produktionsraten zu erreichen und eine möglichst komplikationslose Hochlaufphase zu gewährleisten. [1]. Insbesondere in der Strukturmontage von Rumpfsektionen mit zahlreichen Löchern, die präzises

Bohren und Nieten erfordern, ist der Bedarf an sorgfältiger Planung und fortschrittlichen Automatisierungssystemen hoch [2], [3]. Speziell diese Automatisierungssysteme weisen oft mehrstufige Hierarchien auf, was die Komplexität und damit auch die Wahrscheinlichkeit von Softwareproblemen erhöht [4]. Inmitten dieses Umfeldes wird die Bedeutung der VIBN deutlich. VIBN ist eine Technologie zur Reduzierung von Hochlaufzeiten durch Parallelisierung von Entwicklungs- und Inbetriebnahmeprozessen [1]. Die speziell für bestimmte Anwendungen entwickelte VIBN kann Fehler während der Inbetriebnahme signifikant reduzieren, indem eine gründliche Analyse des SPS-Codes und des Ressourcenverhaltens mithilfe von 3D-Modellen und Steuerungsimulationen vor der eigentlichen Implementierung ermöglicht wird [1]. Obwohl Methoden zur Entwicklung von VIBN-Architekturen existieren, sind sie in der Regel domänen- oder anwendungsspezifisch und berücksichtigen nicht die vielfältigen Anforderungen der Flugzeugfertigung [5]. Die Modellbasierte Systementwicklung (Model-based Systems Engineering, MBSE) bietet wertvolle Methoden und Ansätze zur Entwicklung komplexer Systeme und legt damit den Grundstein für die systematische Entwicklung von VIBN-Architekturen. Ein bekannter Ansatz aus dem Bereich der MBSE ist die SPES-Methodik, die einen systematischen, Hardware- und Softwarekomponenten umfassenden Lösungsansatz für eingebettete Systeme bietet [6]. Aus diesen Beobachtungen ergibt sich das Forschungsziel, zu untersuchen, wie die SPES-Methodik die Entwicklung einer VIBN-Architektur für komplexe Automatisierungssysteme effektiv unterstützen kann. Dieser Beitrag ist wie folgt gegliedert. Im Abschnitt II werden die Grundlagen von VIBN und SPES vorgestellt. Im Abschnitt III wird die VIBN-Architektur systematisch mit Hilfe der SPES-Methodik entwickelt, die die Problembeschreibung des Anwendungsfalls, die Definition der funktionalen und nicht-funktionalen Anforderungen sowie

eine Beschreibung der abgeleiteten Architektur umfasst. Die entwickelte VIBN-Architektur und die Anwendbarkeit von SPES werden im Abschnitt IV diskutiert, während im Abschnitt V die Ergebnisse zusammengefasst und mögliche neue Forschungsfragen aufgezeigt werden.

II. GRUNDLAGEN

A. Grundlagen der virtuellen Inbetriebnahme

Der Inbetriebnahmeprozess stellt sicher, dass ein System, eine Komponente oder eine Anlage entsprechend den betrieblichen Anforderungen geplant, installiert, getestet, betrieben und gewartet wird. Dieser Prozess beginnt in der Regel, sobald das System vor Ort installiert ist. Abbildung 1 vergleicht die reale Inbetriebnahme mit der VIBN für Systemlieferanten und Betreiber. Der Prozess, der VIBN ist ein fortschrittlicher simulationsbasierter Ansatz, der darauf abzielt, den Inbetriebnahmeprozess zu verbessern und bereits frühzeitig im virtuellen Raum gestartet werden kann. Bei der VIBN kommt eine virtuelle Testumgebung hinzu, die Simulationsmodelle und virtuelle Tests beinhaltet. Durch die Verknüpfung von Steuerungs-, Bauteil- und Maschinenmodellen ermöglicht die VIBN eine frühzeitige Überprüfung und Optimierung des Anlagenverhaltens. Die VIBN kann die eigentliche Inbetriebnahme zwar nicht vollständig ersetzen, hilft aber bei der frühzeitigen Erkennung und Behebung von Fehlern, verkürzt die Inbetriebnahmezeit, erhöht die Systemqualität und minimiert die mit der Inbetriebnahme vor Ort verbundenen Kosten. Darüber hinaus kann die VIBN die Kommunikation und Koordination zwischen Lieferanten und Betreibern verbessern, was eine schnellere Umsetzung von Änderungen und fortgeschrittenen Schulungsszenarien ermöglicht [7].

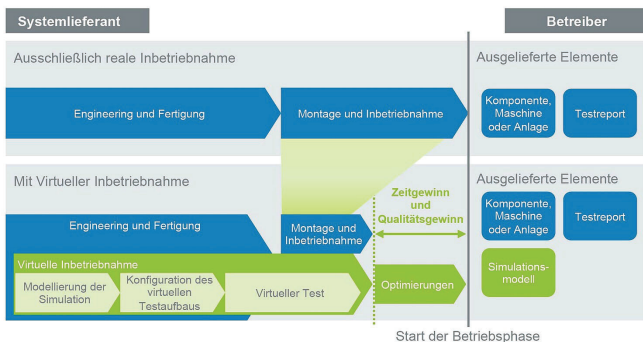


Abbildung 1: Vergleich von realer Inbetriebnahme und VIBN [7]

Der Einsatz von VIBN spielt eine entscheidende Rolle bei der Bewältigung der hohen Komplexität moderner Produktionsanlagen. Diese müssen unter sorgfältiger Berücksichtigung der vorherrschenden Anforderungen entwickelt werden, was die Definition und Charakterisierung wesentlicher Komponenten für die Architektur erfordert.

Die Integration von Verhaltensmodellen, z.B. durch eine Co-Simulationsumgebung, in den VIBN-Workflow ist entscheidend für die Validierung von Steuerungsprogrammen,

durch die umfassende Tests und Fehlerbehebung ermöglicht werden. Auf diese Weise werden die Kommunikationsstruktur und Interoperabilität vor der eigentlichen Inbetriebnahme getestet [8]. Technologien wie ein Functional Mock-Up Interface (FMI) erleichtern eine nahtlose Co-Simulation und AutomationML verbessert den Austausch von Engineering-Daten [9]. Beides ist wichtig im Validierungsprozess. Dazu gehören auch die Integration von Sicherheitsfunktionen, um Systemausfallzeiten zu reduzieren und umfassende Mitarbeiterschulungen [10]. Standard-Technologien, die für Konnektivität und Kommunikation innerhalb von VIBN-Architekturen eingesetzt werden, umfassen Technologien wie OPC UA, TCP/IP oder FMI.

Die Entwicklung einer VIBN-Architektur umfasst mehrere Komponenten. Komponenten wie das Verhaltensmodell, das erweiterte Geometriemodell und die Bus-Emulation ermöglichen die Validierung wichtiger Funktionalitäten, einschließlich der Simulation von Schnittstellen, der internen Verhaltenslogik, der Visualisierung des Systemmodells und der Kommunikation mit realen SPSEN [11]. Virtuelle Darstellungen physischer Objekte, die für VIBN und Digital Twins [12], [13] unerlässlich sind, spielen eine wichtige Rolle bei Simulationen und Tests und verringern die Abhängigkeit von physischen Tests.

Die Entwicklung einer VIBN-Architektur erfordert jedes Mal einen individuellen Ansatz, der auf den spezifischen Anwendungsfall zugeschnitten ist. Faktoren wie die Verfügbarkeit der Emulation des Anlagenverhaltens, die Auswahl der Software, Ressourcenauswahl und Budgetbeschränkungen beeinflussen das Design der VIBN-Architektur. Abhängig davon, ob die Software vom gleichen Hersteller oder von verschiedenen Herstellern stammt, besteht die Notwendigkeit von Co-Simulationsmethoden wie FMIs und Datenschnittstellen [8]. Die Integration des Steuerungssystems in ein Simulationsmodell umfasst im VIBN-Kontext verschiedene Konfigurationen und Methoden [14]. Diese Konfigurationen unterscheiden sich je nach Faktoren wie Typ und Umfang der Modelle, Varianten der Simulationswerkzeuge, Kommunikationsform zwischen der Simulation und den Automatisierungssystemen sowie Typ und Umfang der simulierten Automatisierungssystemstrukturen. Folglich ist die Wahl der VIBN-Setup-Konfiguration, wie Model-in-the-Loop (MiL), Hardware-in-the-Loop (HiL), Software-in-the-Loop (SiL) oder Reality-in-the-Loop (RiL), eng mit den Anforderungen, der Notwendigkeit und der Verfügbarkeit der Elemente des Problems verbunden.

B. Grundlagen der Modellierung nach SPES

Das SPES-Modellierungsframework zielt darauf ab, sprach- und werkzeugunabhängige Modellierungsmethoden für den Entwurf eingebetteter Systeme bereitzustellen [15]. Dieses generische MBSE-Framework findet Anwendung in verschiedenen Domänen. Das Kernkonzept definiert zwei Hauptansätze: Abstraktionsebenen und Sichten, die zusammen einen zweidimensionalen Engineering-Raum bilden [15].

Auf der horizontalen Achse werden die Sichten in vier Abschnitte unterteilt: *Anforderungs-*, *Funktionale-*, *Logische-*

und *Technische* Sicht. Die Anforderungssicht ermöglicht es, den Prozess der Erfassung, Dokumentation und Validierung von Anforderungen zu gestalten. Die Anforderungssicht umfasst die Definition des Kontextmodells des Systems, des Zielmodells, der Szenariomodelle (Use-Cases) und des lösungsorientierten Anforderungsmodells. Die *Funktionale Sicht* umfasst Blackbox- und Whitebox-Modelle und konzentriert sich auf die Funktionalitäten des Systems. In der Blackbox wird die funktionale Struktur auf hoher Ebene beschrieben und iterativ in feinere Funktionen zerlegt. Die Whitebox untersucht das Verhalten der Funktionen und berücksichtigt das Ablaufverhalten sowie Eingaben, Ausgaben und Abhängigkeiten zwischen den Funktionen. Die *Logische Sicht* beschreibt die strukturelle Zusammensetzung des Systems, indem das System of Interest (SoI) in logische Komponenten zerlegt wird, die in der Lage sind, die definierten Funktionalitäten auszuführen. Sie ordnet Funktionen den logischen Komponenten zu, definiert Schnittstellen zwischen den Komponenten und beschreibt die Interaktion und das Verhalten von Teilsystemen während der Ausführung von Funktionen. Schließlich konzentriert sich die *Technische Sicht* auf technische Spezifikationen und verfeinert die in der Logischen Sicht modellierten Elemente. Sie definiert Systeme, Teilsysteme und analysiert Systemvarianten hinsichtlich ihrer Funktionalität und Eignung. Die vertikale Achse repräsentiert verschiedene Abstraktionsebenen, die den spezifischen Systemebenen entsprechen. Sie ermöglicht die Modellierung eines vollständigen Systems oder nur eines seiner Designelemente [15].

III. ANGEPASSTER SPES-ABLAUF FÜR DIE ENTWICKLUNG VON VIBN-ARCHITEKTUREN

Der verwendete Modellierungs-Workflow, der in [16] veröffentlicht wurde, folgt der SPES-Methodik, die im vorherigen Abschnitt beschrieben wurde. Der Workflow umfasst die Entwicklung der vier Sichten und wurde mit Hilfe des Softwaretools Magic Systems of Systems Architect (MSoSA) in der Systems Modeling Language (SysML) implementiert.

A. Anwendungsbeispiel

Eines der Hauptziele bei der Anwendung des angepassten SPES-Workflows besteht darin, die Effizienz bei der Entwicklung von VIBN-Architekturen zu steigern und ihn für verschiedene Anwendungsfälle komplexer automatisierter Systeme zu nutzen. Einer dieser Anwendungsfälle ist das iMOD-Projekt, dessen Ziel es ist, die Effizienz, Qualität und Anpassungsfähigkeit der Flugzeugfertigung zu verbessern. Erreicht werden soll dies durch die Einführung eines speziellen Automatisierungssystems für das Bohren und Nieten in der Rumpfmontage sowie durch neue Engineeringansätze [2], [16].

Im Rahmen des iMOD-Projekts wird eine prototypische Montagesation für die Serienproduktion zukünftiger Flugzeugmodelle aufgebaut. Diese dient als Validierungsumgebung für eine hochautomatisierte Lösung, in der Bohr- und Nietprozesse für die Montage von Flugzeugrümpfen getestet werden. Das automatisierte Montagesystem ist in der Lage,

sowohl die Längs- als auch die Quernaht herzustellen. Die komplexen Prozesse werden von mehreren individuellen Robotern mit multifunktionalen Endeffektoren ausgeführt, die über eine dezentrale Steuerung verfügen und eigenständig Entscheidungen treffen können. Das iMOD-Projekt umfasst auch die Integration wesentlicher IT/OT-Ausrüstung zur Steuerung der automatisierten Komponenten sowie die Implementierung erforderlicher Sicherheitsmaßnahmen für die Plattform.

Die Komplexität des beschriebenen Automatisierungssystems erfordert eine gründliche Validierung vor der eigentlichen Inbetriebnahme. Aufgrund der anspruchsvollen, eng getakteten Inbetriebnahmepläne ist der Einsatz von VIBN daher unumgänglich. Die Entwicklung einer geeigneten VIBN-Architektur ist jedoch eine aufwändige und komplexe Aufgabe, da es keine allgemeingültige Lösung gibt. Sie muss individuell an die Bedürfnisse und Anforderungen der Anlage angepasst werden. Im folgenden Abschnitt wird daher zunächst die Anforderungssicht modelliert, um die notwendigen Anforderungen an die entsprechende VIBN-Architektur zu identifizieren.

B. Anforderungssicht

Um alle funktionalen Anforderungen (FA) und nicht-funktionalen Anforderungen (NFA) für die VIBN-Architektur zu identifizieren, schlägt die SPES-Methodik eine Reihe von Schritten vor, einschließlich der Definition des Systemkontexts und der Definition von Anwendungsfällen.

Der Systemkontext wird mit Hilfe von SysML Internal Block Diagrams (ibd) modelliert, siehe Abbildung 2. Dieses Diagramm bietet einen umfassenden Überblick über die zu berücksichtigenden Kontextelemente und ihre Interaktion mit dem zu entwickelnden VIBN-System. Es umfasst externe Systemkomponenten, wie das *Industrial Network*, und Stakeholder, die über eine definierbare Schnittstelle mit dem System interagieren. Der Systemkontext dient als Grundlage für die Ableitung von NFA. Das VIBN muss in das *Industrial Network*, genauer gesagt in eine Product Lifecycle Management (PLM)-Umgebung, integriert werden, um die Kompatibilität mit der etablierten industriellen IT-Infrastruktur zu gewährleisten. Die *Station* verfügt über eine bidirektionale Echtzeitkommunikation mit dem SoI, die die Interoperabilität mit den Steuerungssystemen und den realen Geräten ermöglicht. Aus diesem Modell lassen sich die Anforderungen *NFA4* und *NFA5* ableiten.

Die Stakeholder *Expert User*, *PLC Programmer* und *Offline Programmer* müssen ebenfalls betrachtet werden. Um eine übersichtliche Darstellung zu gewährleisten, wird darauf verzichtet weitere Stakeholder wie den *Bediener* und *Wartungs-Mitarbeiter* in der Abbildung darzustellen. Der *Offline Programmer* muss in der Lage sein, Robotermodelle hinzuzufügen, zu ändern oder zu entfernen. Darüber hinaus müssen sowohl der *Expert User* als auch der *PLC Programmer* die Möglichkeit besitzen Hardware- und Softwaremodule hinzuzufügen, zu ändern oder zu entfernen. Daraus ergibt sich die Anforderung *NFA1* eines modularen Designs und der Skalierbarkeit der Architektur. Darüber hinaus benötigen

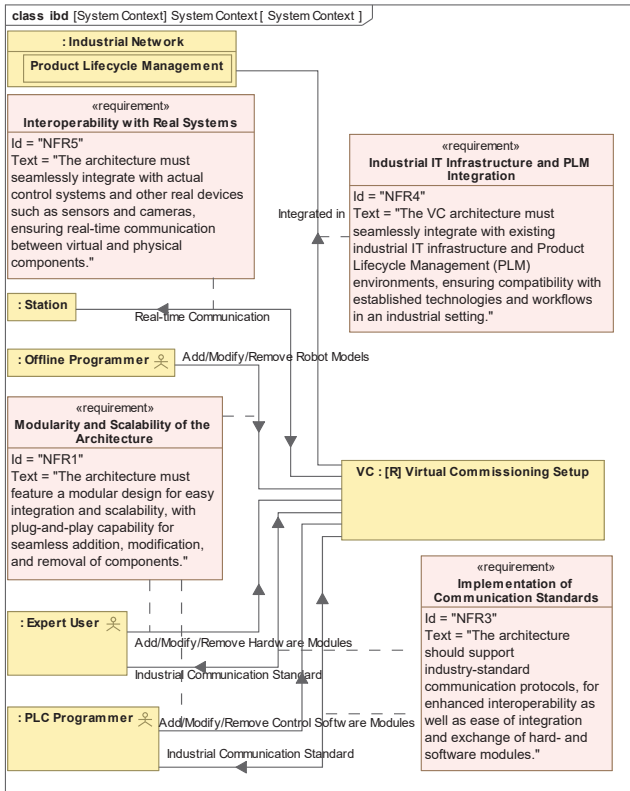


Abbildung 2: Anforderungssicht - Systemkontext [17]

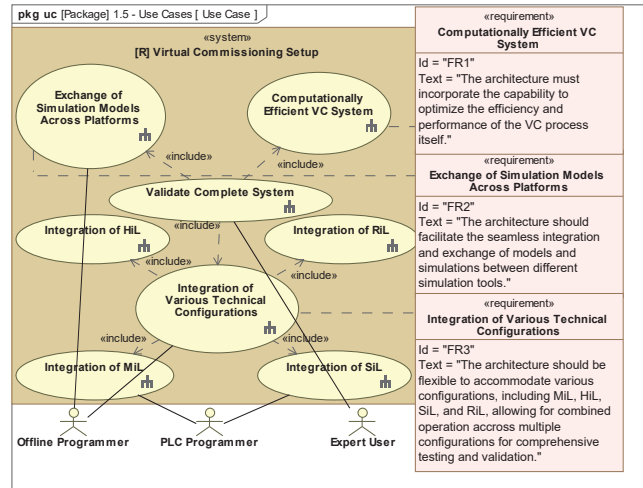


Abbildung 3: Anforderungssicht - Anwendungsfalldiagramm [17]

sowohl der *Expert User* als auch der *PLC Programmer* bidirektionale Kommunikationsschnittstellen, welche auf industriellen Kommunikationsstandards basieren. Die Verwendung von industrietypischen Kommunikationsprotokollen verbessert die Interoperabilität und ermöglicht die Integration und den Austausch von Hard- und Softwaremodulen. Dies ist in *NFA3* festgehalten.

Die Anforderung *NFA2* (Adaptive und dezentrale Steuerungsarchitektur) ist im System Context Diagramm nicht explizit dargestellt, ergibt sich aber aus der Notwendigkeit einer Steuerungsarchitektur, die eine flexible Anpassung und Integration von Modulen und Komponenten zur Laufzeit ermöglicht. Dies ist besonders wichtig, da einige der hinzugefügten Komponenten teilweise ihre eigene Steuerungsintelligenz haben, aber nahtlos in das System integriert werden müssen. Das Ziel ist eine dezentrale Architektur mit hierarchischen und dezentralen Funktionen, um ein effizientes Systemmanagement zu gewährleisten.

Abbildung 3 stellt die Anwendungsfälle des Systems in einem SysML Use Case Diagramm dar. Der Hauptanwendungsfall, *Validate Complete System*, umfasst verschiedene Unteranwendungsfälle. Der Anwendungsfall *Computationally Efficient VC System* zielt darauf ab, die Effizienz und Leistung des VIBN-Setups zu optimieren. Von diesem wird die Anforderung *FA1* abgeleitet. Der Anwendungsfall *Exchange of Simulation Models Across Platforms* bezieht sich

auf die Notwendigkeit einer nahtlosen Integration und des Austauschs von Modellen zwischen verschiedenen Simulationswerkzeugen und resultiert in *FA2*. Der Anwendungsfall *Integration of Various Technical Configurations* zielt darauf ab, die Flexibilität bei der Aufnahme verschiedener technischer Konfigurationen wie MiL, HiL, SiL und RiL zu gewährleisten und führt zu *FA3*. Das Diagramm hebt auch die Stakeholder hervor, die an diesen Anwendungsfällen beteiligt sind.

Funktionale und nicht-funktionale Anforderungen - Im Zuge der Entwicklung einer VIBN-Architektur für den Anwendungsfall iMOD wurden der System Context (vgl. Abbildung 2) und die Use Cases (vgl. Abbildung 3) modelliert und daraus die funktionalen und nicht-funktionalen Anforderungen abgeleitet. Auch wenn diese Anforderungen nur einen Teil der gesamten Anforderungen umfassen, dienen speziell diese als Referenzpunkt für die Bewertung und dem Vergleich mit bestehenden vorhandenen Architekturen in der Literatur. Diese Anforderungen sind so vielseitig, sodass die entsprechende VIBN-Architektur auch auf anderen Anwendungsfälle anwendbar ist, die über die iMOD Station hinausgeht.

- NFA1 *Modularität und Skalierbarkeit der Architektur*
- NFA2 *Anpassungsfähige und dezentralisierte Steuerungsarchitektur*
- NFA3 *Implementierung von Kommunikationsstandards*
- NFA4 *Industrielle IT-Infrastruktur und PLM-Integration*
- NFA5 *Interoperabilität mit realen Systemen*
- FA1 *Recheneffizientes VIBN-System*
- FA2 *Plattformübergreifender Austausch von Simulationsmodellen*
- FA3 *Integration von verschiedenen technischen Konfigurationen*

Analyse bestehender VIBN-Setups - Bevor die funktionale Sicht definiert wurde, wurden bestehende VIBN-Setups aus der Literatur hinsichtlich ihrer Eignung analysiert. Dazu wurde ein Abgleich der definierten Anforderungen mit den

bestehenden Ansätzen durchgeführt und die Ergebnisse in Tabelle I zusammengefasst. Jeder Ansatz präsentiert seine eigenen Zielsetzungen und spezifischen Anwendungsbereiche, die das Design und den Fokus der jeweiligen Architekturen beeinflussen. Der Vergleich ergab, dass zwar einige Architekturen bestimmte Anforderungen effektiv adressierten, jedoch keiner der Ansätze alle definierten Anforderungen erfüllt. Daraus ergibt sich die Notwendigkeit, einen neuen VIBN-Architekturansatz zu entwickeln, der diese generischen Anforderungen erfüllt und damit sowohl im Kontext des iMOD-Anwendungsfalles als auch in anderen Anwendungsfällen eingesetzt werden kann.

Tabelle I: Vergleich von VIBN-Architektur Ansätzen [17]

Referenz	NFA1	NFA2	NFA3	NFA4	NFA5	FA1	FA2	FA3
Makris et al. [18]	1	1	1	0	0	0	0	0
Ko et al. [19]	1	0	1	0	1	0	0	1
Stieß et al. [9]	1	1	1	0	0	0	1	1
Vermaak and Niemann [20]	1	0	1	0	1	0	0	1
Hofmann et al. [21]	1	0	1	0	1	0	0	1
Illmer et al. [22]	1	0	1	0	0	0	0	0
Barbieri et al. [13]	1	0	1	0	0	0	0	0
Abassi et al. [10]	1	0	1	0	0	0	0	0
Bendjelloul et al. [23]	1	1	1	0	1	0	0	1
Raffaelli et al. [24]	1	0	1	0	1	0	0	1
Ismail and Fay [3]	0	0	1	0	0	1	0	0

Legende: 1 = erfüllt Anforderung, 0 = berücksichtigt Anforderung nicht

C. Funktionssicht

Die funktionale Sicht vertieft die Umsetzung der identifizierten Anwendungsfälle. In der Blackbox-Perspektive werden Funktionen zunächst in einem Blockdefinitionsdiagramm definiert. Übergeordnete Systemfunktionen werden zunächst zu Funktionen, die dann iterativ in granulare Funktionen, sogenannte atomare Blackbox-Funktionen, zerlegt werden.

In einem zweiten Schritt wird innerhalb der Whitebox-Perspektive der zeitliche und logische Ablauf der Systemfunktionen beschrieben (vgl. Abbildung 4). Der Anwendungsfall *Austausch von Simulationsmodellen zwischen Plattformen* aus Abbildung 3 wird hier in vier atomare Blackbox-Funktionen zerlegt. Die erste Funktion generiert das zu untersuchende Modell, die zweite Funktion testet das Steuerungsverhalten des Modells, die dritte realisiert den Austausch und die Integration von Modellen zwischen verschiedenen Simulationswerkzeugen, und die vierte testet das Modell in der 3D-Simulationssoftware.

Die Modellierung der funktionalen Sicht ist ein iterativer Prozess. So ist es möglich, dass innerhalb der Whitebox-Perspektive übersehene Funktionen identifiziert werden. Diese müssen nachträglich modelliert werden. Außerdem hat dies direkten Einfluss auf die Modellierung der Blackbox-Perspektive da dort ebenfalls vergessene Funktionen ergänzt werden müssen. Dieser dualistische Ansatz, der sowohl die Blackbox- als auch die Whitebox-Perspektiven umfasst, trägt dazu bei, sowohl die vollständigen Funktionen als auch die

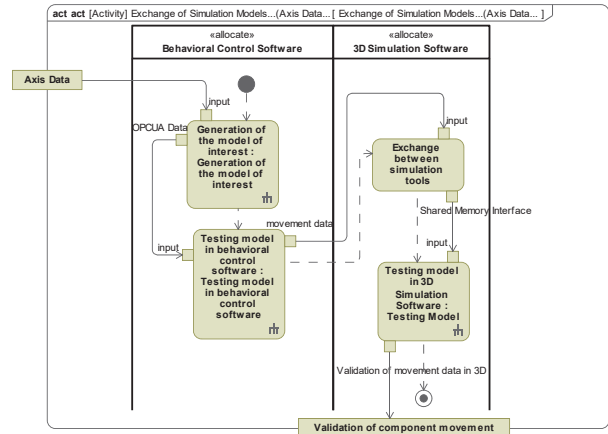


Abbildung 4: Funktionssicht - Whitebox-Perspektive [17]

komplexen Details des Verhaltens des Systems zu definieren und somit zur umfassenden Entwicklung der VIBN-Architektur beizutragen.

D. Logische Sicht

Die logische Sicht beginnt mit der Definition der logischen Subsysteme in einem Blockdiagramm. Ausgehend von dem zu entwickelnden SoI werden die Teilkomponenten definiert, die zur Erfüllung der Systemfunktionen benötigt werden. In einem zweiten Schritt werden mit Hilfe von Interfaceblöcken, die Interaktionen zwischen den Komponenten definiert vgl. Abbildung 5. Diese werden unter anderem auch aus den Schnittstellen zwischen dem System und dem System Context sowie den funktionalen Schnittstellen abgeleitet.

Nach der Modellierung der funktionalen und der logischen Sicht werden diese miteinander verknüpft, indem jede Funktion einer logischen Komponente zugeordnet wird. Im Beispiel des *Exchange of Simulation Models Across Platforms* und seiner vier Funktionen, dargestellt in Abbildung 4, sind für jede Funktion spezifische Subsysteme vorgesehen, z.B. wird für die Generierung und den Test des Modells eine Verhaltenssteuerungssoftware benötigt und für den Test in einer 3D-Umgebung zusätzlich eine 3D-Simulationssoftware. Damit Daten zwischen diesen zwei Software-Werkzeugen ausgetauscht werden können, bedarf es weiterhin einer logischen Schnittstelle zwischen den entsprechenden logischen Komponenten. Auf die gleiche Weise wird auch jede andere Funktion einer logischen Komponente zugewiesen. Die Definition der benötigten Teilsysteme sowie ihrer Schnittstellen ist dabei stark von der Expertise und Erfahrung des Entwicklers abhängig. Das Übersehen von Komponenten oder Schnittstellen in der logischen Sicht führt zu einem iterativen Korrekturprozess, der bis in die Anforderungssicht zurückreichen kann.

E. Technische Sicht

Schließlich wurde in der technischen Sicht die eigentliche VIBN-Architektur definiert. Basierend auf den logischen

DANKSAGUNG

Diese Forschungsarbeit aus dem Projekt iMOD wird durch dtec.bw – Zentrum für Digitalisierungs- und Technologieforschung der Bundeswehr gefördert. dtec.bw wird von der Europäischen Union – NextGenerationEU finanziert.

LITERATUR

- [1] A. Jain, D. A. Vera, and R. Harrison, "Virtual Commissioning of Modular Automation Systems," *IFAC Proceedings Volumes*, vol. 43, no. 4, pp. 72–77, 2010.
- [2] F. Gehlhoff, H. Nabizada, M. Weigand, L. Beers, O. Ismail, A. Wenzel, A. Fay, P. Nyhuis, W. Lagutin, and M. Röhrig, "Challenges in Automated Commercial Aircraft Production," *IFAC-PapersOnLine*, vol. 55, no. 2, pp. 354–359, 2022.
- [3] O. Ismail and A. Fay, "Optimizing virtual commissioning of a robotic system using process mining and footprints conformance checking," in *2023 IEEE 21st INDIN*, 2023, pp. 1–6.
- [4] B. Vogel-Heuser, G. Kegel, K. Bender, and K. Wucherer, "Global information architecture for industrial automation," *Automatisierungstechnische Praxis (atp)*, vol. 51, no. 1, pp. 108–115, 2009.
- [5] N. H. Shah, P. Le Henaff, C. Schiffer, M. Krammer, and M. Benedikt, "Accurate robot simulation for industrial manufacturing processes using fmi and dcp standards," in *Proceedings of 14th Modelica Conferences*, 2021.
- [6] K. Pohl, M. Broy, H. Daembkes, and H. Hönninger, *Advanced Model-Based Engineering of Embedded Systems*. Springer, 2016.
- [7] H. Rauhen, C. Mosch, and E. Axmann, "VDMA Virtual Commissioning Guide: Recommendations for action for economic entry," 2020.
- [8] S. Süß, A. Strahilov, and C. Diedrich, "Behaviour simulation for virtual commissioning using co-simulation," in *2015 IEEE 20th ETFA*, 2015, pp. 1–8.
- [9] S. Süß, S. Magnus, M. Thron, H. Zipper, U. Odefey, V. Fäßler, A. Strahilov, A. Klodowski, T. Bär, and C. Diedrich, "Test methodology for virtual commissioning based on behaviour simulation of production systems," in *2016 IEEE 21st ETFA*, 2016, pp. 1–9.
- [10] A. Abassi, J. H. Lugo-Calles, R. Balderas-Hill, and N. Lassabe, "Virtual Commissioning Implementation of Industrial Painting Process," in *2nd International Conference CompAuto*. IEEE, 2022, pp. 65–68.
- [11] Z. Liu and C. Diedrich, "Component modeling for different application scenarios in the virtual commissioning," in *2020 IEEE 7th ICISCE*, 2020, pp. 1774–1778.
- [12] D. Rovere, M. Silvestri, G. D. Maso, H. Dzafic, and P. Pedrazzoli, "Streamline 3D simulation model development for virtual commissioning with IEC61499," *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 773–778, 2022.
- [13] G. Barbieri, A. Bertuzzi, A. Capriotti, L. Ragazzini, D. Gutierrez, E. Negri, and L. Fumagalli, "A virtual commissioning based methodology to integrate digital twins into manufacturing systems," *Production Engineering*, vol. 15, no. 3-4, pp. 397–412, 2021.
- [14] N. Striffler and T. Voigt, "Concepts and trends of virtual commissioning – A comprehensive review," *Journal of Manufacturing Systems*, vol. 71, pp. 664–680, 2023.
- [15] K. Pohl, H. Hönninger, R. Achatz, and M. Broy, *Model-Based Engineering of Embedded Systems*. Springer, 2012.
- [16] L. Beers, M. Weigand, H. Nabizada, and A. Fay, "MBSE Modeling Workflow for the Development of Automated Aircraft Production Systems," in *2023 IEEE 28th ETFA*, 2023, pp. 1–8.
- [17] O. Ismail, L. Beers, F. Gehlhoff, N. H. Shah, and A. Fay, "Systematic development of a virtual commissioning architecture for an automated production system," in *2024 IFAC 18th INCOM*, 2024, accepted for publication.
- [18] S. Makris, G. Michalos, and G. Chryssolouris, "Virtual commissioning of an assembly cell with cooperating robots," *Advances in Decision Sciences*, vol. 2012, 2012.
- [19] M. Ko, E. Ahn, and S. C. Park, "A concurrent design methodology of a production system for virtual commissioning," *Concurrent Engineering*, vol. 21, no. 2, pp. 129–140, 2013.
- [20] H. Vermaak and J. Niemann, "Virtual commissioning: A tool to ensure effective system integration," in *2017 IEEE ECMSM*, 2017, pp. 1–6.
- [21] W. Hofmann, J. H. Ulrich, S. Lang, T. Reggelin, and J. Tolujew, "Simulation and Virtual Commissioning of Modules for a Plug-and-Play Conveying System," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 649–654, 2018.
- [22] B. Illmer, M. Karkowski, and M. Vielhaber, "Petri net controlled virtual commissioning – A virtual design-loop approach," *Procedia CIRP*, vol. 91, pp. 152–157, 2020.
- [23] A. Bendjelloul, B. Mihoubi, M. Gaham, M. Moufid, and B. Bouzouia, "A framework for an effective virtual commissioning of agent-based cyber-physical production systems integrated into manufacturing facilities," *Concurrent Engineering*, vol. 30, no. 4, pp. 399–410, 2022.
- [24] R. Raffaelli, P. Bilancia, F. Neri, M. Peruzzini, and M. Pellicciari, "Engineering Method and Tool for the Complete Virtual Commissioning of Robotic Cells," *Applied Sciences*, vol. 12, no. 6, p. 3164, 2022.