18. AALE-Konferenz
Pforzheim, 09.03.-11.03.2022

Virtualisierungsarchitektur für heterogene CAx-Tools im Engineering modularer Produktionssysteme

G. Hildebrandt[1], P. Habiger[2], R. Drath[3], M. Barth[4]

Zusammenfassung

Modularisierung und virtuelles Engineering sind vielversprechende Lösungsansätze zur Bewältigung aktueller Herausforderungen in der industriellen Produktion. Vor dem Hintergrund der Produktindividualisierung werden monolithische Produktionssysteme zunehmend durch flexibel konfigurierbare Anlagen ersetzt. Diese bestehen aus wiederverwendbaren und für unterschiedliche Anwendungsszenarien kombinierbaren Modulen. Als Resultat dieser Entwicklung werden die entsprechenden Module zu evolvierenden Engineering-Lösungen, deren Versionen verwaltet werden müssen. Weiterhin sollen im Modulengineering entwickelte (Simulations-)Modelle möglichst flexibel wiederverwendet werden, was sich aufgrund der großen Zahl an proprietären Entwicklungswerkzeugen (CAx-Tools) als schwierig erweist. Im Rahmen dieser Arbeit stellen die Autoren eine Architektur vor, die ein effizientes, virtuelles und modulares Engineering ermöglicht. Mit Hilfe von Virtualisierungsmethoden wird die Wiederverwendung beliebiger, heterogener Modelle und Simulationen durch die Weitergabe der zugehörigen CAx-Tools möglich. Das vorgestellte Konzept ist prototypisch umgesetzt und zeigt damit dessen Machbarkeit auf. Abschließend werden Kritikpunkte der Architektur sowie mögliche Lösungsansätze erörtert und bestehende Forschungsfragen aufgezeigt.

Stichwörter

Virtuelles Engineering, Modularisierung, Verwaltungsschale, Virtualisierung

1 Motivation und Problemstellung

Die industrielle Produktion befindet sich, bedingt durch die Nachfrageverschiebungen, inmitten eines Wandels, weg von günstigen Massenartikeln hin zu individuell zugeschnittenen Produkten. Zeitgleich verkürzen sich Produktlebenszyklen, was für immer kürzere Produktentwicklungszeiträume sorgt [1]. Beide Aspekte wirken sich auch auf die produzierenden Gewerbe und verwendeten Produktionsanlagen aus. Moderne Produktionssysteme müssen flexibel sein und sich schnell an ändernde Produkt- und Produktionsanforderungen anpassen – etwa Aufträge der Losgröße 1 [2]. Als Reaktion, auf die sich ändernden Anforderungen werden aktuell mehrere Ansätze diskutiert. Zwei vielversprechende Lösungen sind die Modularisierung und das virtuelle Engineering von Produktionsanlagen.
Unter Modularisierung wird im Anlagen- und Maschinenbau die Auftrennung komplexer Produktionsprozesse in grundlegende Funktionseinheiten verstanden. Die resultierenden Module können grundlegende Prozesse ausführen und auf Basis standardisierter Schnittstellen zu Gesamtanlagen

[1] Gary Hildebrandt, Hochschule Pforzheim, Wissenschaftlicher Mitarbeiter
[2] Pascal Habiger, Hochschule Pforzheim, Wissenschaftlicher Mitarbeiter
[3] Prof. Dr.-Ing. Rainer Drath, Hochschule Pforzheim, Prof. Mechatronische Systementwicklung
[4] Prof. Dr.-Ing. Mike Barth, Hochschule Pforzheim, Prof. Engineering mechatronischer Komponenten

kombiniert werden. Wesentliche Vorteile von Modularisierung sind somit die Wiederverwendung von Modulen in unterschiedlichen Produktionsszenarien, die Austauschbarkeit der Module und die bessere Skalierbarkeit modularer Anlagen [3]. Der zweite Ansatz, das virtuelle Engineering, beschreibt den Einsatz von simulativen/modellbasierten Methoden über den gesamten Engineeriungzyklus [4]. Dies wird durch die möglichst vollständige Digitalisierung von Engineering-Artefakten (EA), wie Simulationsmodellen, Datenblättern, Bedienungsanleitungen, Steuerungscode, Bauteilzeichnungen, 3D-Geometriemodellen und vielem mehr, ermöglicht und erleichtert [5].
Die Kombination der beiden Ansätze hat zur Folge, dass die Methoden des virtuellen Engineerings auch auf die Module angewandt werden. Durch die Wiederverwendung und Vermeidung von Neuentwicklungen werden Module zu evolvierenden Engineering-Lösungen, welche – ähnlich zu Software – in Versionen veröffentlicht und verkauft werden. Die erste verkaufsbereite Variante eines Moduls würde somit Version 1 entsprechen, die Weiterentwicklung Version 2 etc. Fehlerbehebungen, Firmware-Updates o.ä. können durch Subversionen (v1.3 oder v2.6) abgebildet werden. Auch Modulspezialisierungen könnten durch den Versionsmechanismus gehandhabt werden. Für ein Transportmodul sind Versionen wie 1.5-Schwerlast oder 1.5-Schnelltransport denkbar. Der beschriebene Mechanismus geht mit einer beachtlichen Menge an zugehörigen Daten einher, welche ebenfalls versioniert und archiviert werden müssen.

Die Umsetzung des modularen virtuellen Engineerings ist ein Teilziel des vom BMBF geförderten Forschungsprojektes METHODS (**M**odular **E**ngineering **T**echniques for **H**eter**O**genous **D**iscrete **S**ystems) [6]. In Bild 1 sind die Teilziele des Projektes zur Umsetzung des Plug&Produce-Prinzips in einer heterogenen, gemischt real-virtuellen Umgebung dargestellt. Insgesamt werden innerhalb des Projektes zwei Arbeitsziele verfolgt. Arbeitsziel 1 beschäftigt sich mit der Umsetzung des modularen, heterogenen Plug&Produce für Fertigungsanlagen. Als Ausgangspunkt dienen hierfür klassische monolithische und herstellergebundene Fertigungsanlagen, welche im ersten Schritt in modulare, herstellergebundene Anlagen überführt werden sollen. Anschließend sollen die Konzepte auch auf Module anderer Hersteller übertragen werden, um ein Engineering für herstellerunabhängige, modulare Anlagen zu ermöglichen. Arbeitsziel 2 umfasst die Erarbeitung von Konzepten für ein gemischt real-virtuelles Engineering modularer Produktionsanlagen. Hierfür soll im ersten Schritt ein Konzept entwickelt werden, welches ein modulares, virtuelles Engineering ermöglicht. Basierend hierauf sollen Möglichkeiten untersucht werden, reale und virtuelle Module zu mischen. Die Kombination der beiden Arbeitsziele führt zu dem Gesamtziel eines modularen, heterogenen und gemischt real-virtuellem Engineering. Die in dieser Arbeit vorgestellte Architektur bezieht sich auf den mit einem roten Stern markierten Aspekt von AZ 2 – der Umsetzung eines modularen, virtuellen Engineerings.

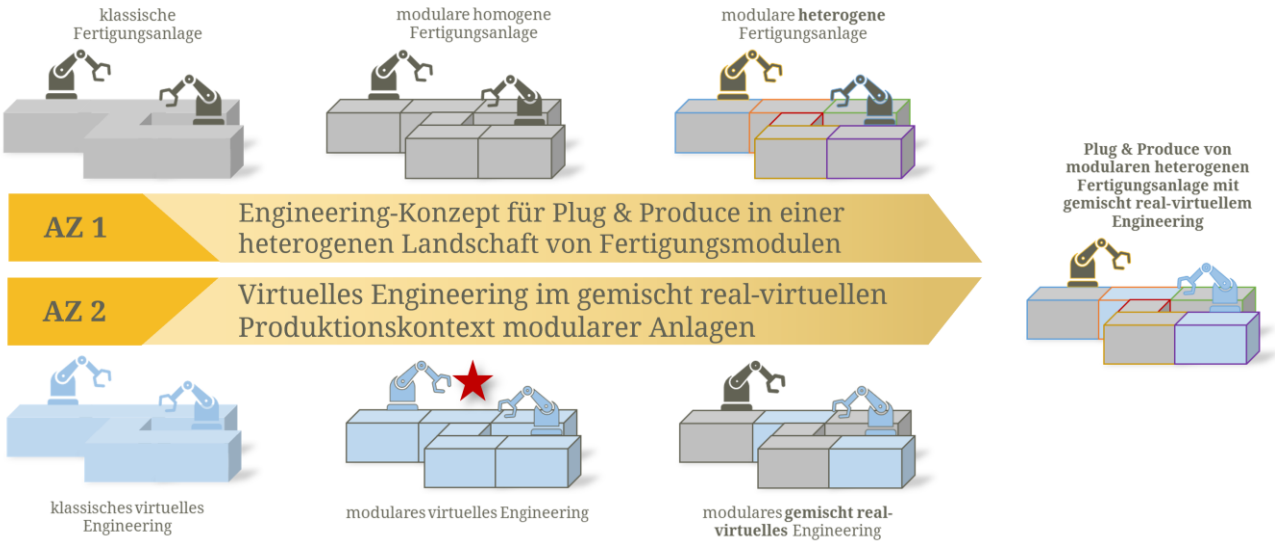

Bild 1 Gesamtzielstellung und Teilziele des METHODS-Projektes

Wie bereits erwähnt, bedient sich das virtuelle Engineering vorwiegend simulationsbasierter Methoden, um potenzielle Engineering-Lösungen virtuell entwickeln, frühzeitig testen und validieren zu können. Im Engineering selbst sind verschiedene Domänen wie mechanisches, elektrisches und automatisierungstechnisches, sowie IT-Engineering enthalten und kombiniert. Aufgrund der unterschiedlichen Anforderungen an die Domänen, unterscheiden sich auch deren Engineering-Werkzeuge (CAx-Tools), was im Laufe der Zeit zu einer Vielzahl von domänenspezifischen CAx-Tools geführt hat. Hieraus resultiert, dass verschiedene Modulhersteller eine für sich funktionierende Tool-Landschaft festgelegt haben, welche jedoch nicht zwangsläufig mit der Tool-Landschaft anderer Hersteller bzw. Kunden kompatibel ist. Während des Modulengineerings erzeugte EA, sollten bestenfalls auch im Anlagenengineering wiederverwendet werden. Dies ist aufgrund der unterschiedlichen Tool-Landschaften von Modulherstellern und Systemintegrator bzw. Anlagenbetreiber oft nicht möglich.

Mit Hilfe der in diesem Beitrag vorgestellten Architektur sollen daher die folgenden Fragestellungen beantwortet werden:

1. Wie kann ein modellbasierter Engineeringprozess für modulare Anlagen unter Berücksichtigung des evolvierenden Charakter von Engineering-Lösungen aussehen?
2. Wie kann es Systemintegratoren und Anlagenbetreibern ermöglicht werde, auf die Engineering-Artefakte von Modulherstellern zuzugreifen, ohne hierfür die CAx-Tools eines jeden Modulherstellers zu erwerben?

Zur Beantwortung dieser Fragen ist der Beitrag wie folgt aufgebaut. Kapitel 2 liefert die Grundlagen der Technologien und Konzepte, welche in der vorzustellenden Architektur verwendet werden. Kapitel 3 analysiert verwandte, relevante Arbeiten aus Wissenschaft und Industrie. In Kapitel 4 wird eine mögliche Architektur-Lösung vorgestellt und ein passendes Anwendungsbeispiel beschrieben. Eine Beschreibung der prototypischen Implementierung des Konzeptes mit den hierfür verwendeten Technologien erfolgt in Kapitel 5. Die Arbeit schließt mit einer Diskussion der vorgestellten Architektur, der Offenlegung noch zu klärender Fragestellungen und einem Ausblick auf folgende Arbeiten.

2 Grundlagen

Die in diesem Beitrag vorgestellte Lösung basiert auf unterschiedlichen Technologien und Konzepten, wie z. B. Virtualisierung, Informationsmodellierung und der I4.0-Verwaltungsschale, deren Grundlagen im Folgenden erläutert werden.

2.1 Virtualisierung

Die Virtualisierung beschreibt ein Konzept, bei dem zwischen zwei Schichten eines Systems eine Abstraktionsebene eingefügt wird, um eine Schicht des Systems (Hard- oder Software) nachzubilden und so der anderen Schicht notwendige Umstände vorzutäuschen [7]. Die Art der Virtualisierung ist dabei vielseitig. Für die vorliegende Arbeit sind zwei Varianten maßgeblich.

2.1.1 Hardware-Virtualisierung

Als Hardware-Virtualisierung wird ein Konzept bezeichnet, bei dem mehreren Komponenten (hier: Betriebssysteme) mit Hilfe einer Abstraktionsebene (hier: Hypervisor) vorgetäuscht wird, sie würden direkt auf benötigte Hardware zugreifen. Tatsächlich wird der Zugriff durch den Hypervisor nur vorgetäuscht und die real zur Verfügung stehenden Ressourcen je nach Bedarf auf die Komponenten aufgeteilt. Dies ermöglicht es beispielsweise mehrere Betriebssysteme auf einer Hardware zu betreiben.

Hauptgründe für diese Art von Virtualisierung sind eine effizientere Auslastung der zugrundeliegenden Hardware und die daraus resultierende Einsparung von Ressourcen. Der Hypervisor verringert durch die Isolation verschiedener Betriebssysteme die Sicherheitsrisiken, wie etwa die Auswirkungen eines Virusbefalls. Wie in Bild 2 dargestellt, können Hypervisoren in Typ-1 und Typ-2 unterschieden werden. Bei Typ-1 Hypervisoren befindet sich die grau dargestellte Abstraktionsschicht zwischen der Hardware und dem Betriebssystem – jedes Betriebssystem befindet sich in diesem Fall auf der gleichen Hierarchieebene. Typ-2 Hypervisoren hingegen werden als Software in einem Wirt-Betriebssystem installiert und dienen als Basis für weitere zu installierende Gast-Betriebssysteme. Jedes Betriebssystem auf einem Hypervisor mit den darin enthaltenen Anwendungen und Dateien, wird als eine virtuelle Maschine (VM) bezeichnet. Die Hardware-Virtualisierung ermöglicht es, virtuelle Maschinen wie ein Datenpaket zwischen verschiedener Hardware auszutauschen und zu betreiben. [7]

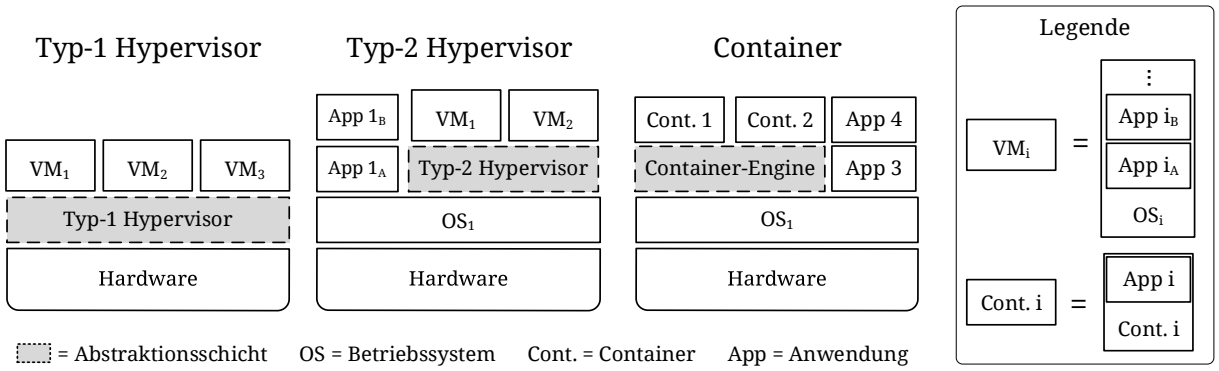

Bild 2 - Unterschiedliche Varianten der Virtualisierung

2.1.2 Betriebssystem-Virtualisierung

Im Vergleich zur Hardware-Virtualisierung findet die Betriebssystem-Virtualisierung auf einer „höheren“ Ebene im Sinne der Computerarchitektur statt. Wie in Bild 2 dargestellt abstrahiert die Betriebssystem-Virtualisierung das Betriebssystem und bildet eine Abstraktionsschicht (hier: Container-Engine) zwischen den Anwendungen und dem Betriebssystem selbst. Ein Container stellt alle notwendigen Abhängigkeiten und Bibliotheken zur Ausführung der Anwendungen bereit, benötigt so keine zusätzliche Installation im Betriebssystem und isoliert die verschiedenen Anwendungen voneinander. Damit wird die Bereitstellung von Anwendungen erleichtert. Gegenüber der Hardware-Virtualisierung bietet diese Variante den Vorteil, dass sie weniger Speicherplatz benötigt. Während eine VM meist mehrere Gigabyte an Speicher einnimmt, genügen bei einem Container teilweise schon wenige Megabyte. Zudem sind Container schneller portier- und verwendbar. Das Erstellen und Hochfahren einer VM kann mehrere Minuten in Anspruch nehmen, während es bei Containern wenige Sekunden sind. Auch eine Kombination von Hardware- und Betriebssystem-Virtualisierung ist möglich.

2.2 Konzept der Verwaltungsschale und des Assets

Einer der wesentlichen Aspekte von Industrie 4.0 ist es, sämtliche Entitäten bzw. Assets innerhalb der Produktion mit einer standardisierten digitalen Beschreibung – der Verwaltungsschale (VWS) – auszustatten. Die VWS beinhaltet oder referenziert auf die digitalen EA des repräsentierten Assets. Die EA werden in Teilmodellen geclustert, welche im Kontext der industriellen Produktion verschiedene Sichten auf die Informationen über das repräsentierte Assets darstellen. Die VWS selbst ist im Netz auffindbar und kann, mit entsprechenden Berechtigungen, nach den gewünschten Inhalten durchsucht werden. Das Konzept der VWS unterscheidet grundlegend zwischen Typen und Instanzen von Assets.

Der Typ kann dabei als Blaupause eines Assets verstanden werden und enthält alle Eigenschaften, welche auch von allen Instanzen des Assets geteilt werden. Eine Instanz beschreibt eine konkrete und eindeutig identifizierbare Einheit eines bestimmten Typs. Entsprechend hierzu existieren somit Typ-Verwaltungsschalen und Instanz-Verwaltungsschalen [8]. Beispielsweise enthält der Typ eines Moduls bereits Informationen über die Abmaße des Moduls, erst in der Instanz können jedoch Eigenschaften wie die Seriennummer des Moduls oder die IP-Adresse der Steuerung angegeben werden. Neben den statischen Informationen können auch dynamische Daten in der VWS enthalten sein. Diese können zur Laufzeit vom Asset oder einer anderen Komponente geschrieben und/oder gelesen werden [9].

3 Relevante Arbeiten

Die Verwendung von Virtualisierungsmethoden innerhalb der industriellen Automatisierung wird bereits in verschiedenen Arbeiten aufgegriffen. So beschreiben BREIVOLD ET AL. [10] vor welchen Herausforderungen die industrielle Automatisierung aktuell und in Zukunft steht, und ob bzw. wie diese mit Hilfe der Virtualisierung gemeistert werden können. Hierbei werden Aspekte der steigenden Konnektivität und damit einhergehende Sicherheitsrisiken, voranschreitende Dezentralisierung der Automatisierungskomponenten und steigende Zahl von zu verarbeitenden Daten und Informationen genannt. Der Fokus von BREIVOLD ET AL. liegt auf der Identifikation potenzieller Anwendungsbereiche der Virtualisierung, nennt aber keine konkreten Konzepte, wie diese in den einzelnen Bereichen umgesetzt werden können. Weiterhin beziehen sich die identifizierten Herausforderungen auf die Betriebsphase der Automatisierungssysteme, jedoch nicht auf deren Engineering.
AZARMIPOUR ET AL. [11] beschreiben konkretere Konzepte zur Anwendung der Virtualisierung in der Betriebsphase und zeigen auf, dass sich mit Hilfe von Hardware-Virtualisierung viele Aspekte eines Automatisierungssystems wie die Prozessführung, das Human-Machine-Interface und die Assetverwaltung auf derselben Hardware ausführen lassen. Durch die Isolation der verschiedenen Partitionen, ist es möglich, relevante Sicherheitsaspekte zu realisieren. Hauptvorteil des dort vorgestellten Konzeptes betrifft die dynamische Allokation von Ressourcen und die Möglichkeit, kontinuierliche Updates durchzuführen, ohne den Betrieb der Automatisierungslösung zu unterbrechen [12].
Anwendungspotenziale von Virtualisierung in der Engineering-Phase werden von BREIVOLD ET AL. in [13] beschrieben. So kann mit Hilfe virtualisierter Testumgebungen eine leichtere Skalierbarkeit, erhöhte Flexibilität und einfachere Nachbildung von Fehlersituationen in Automatisierungssystemen realisiert werden. Es wird beschrieben, welche Bereiche des Engineerings durch Virtualisierung verbessert werden können, jedoch fehlt ein Konzept zur Integration derartiger Methoden in den Engineeringprozess.
Die relevanten Arbeiten fokussieren die Virtualisierung in der Betriebsphase von Systemen und zeigen Verbesserungspotenziale im Engineering auf. Es fehlen jedoch – wie in dieser Arbeit vorgestellt – Konzepte, mithilfe derer die Virtualisierung bereits im Engineering effizient eingesetzt werden kann. Daher überführt die in Abschnitt 4 vorgestellte Architektur die Konzepte der Virtualisierung aus der Betriebsphase von Automatisierungslösungen in deren Engineering und leistet so einen Beitrag dazu, dass die Virtualisierung im gesamten Systemlebenszyklus zum Einsatz kommt.

4 Virtualisierungsarchitektur

Der Kern der Arbeit ist das in Abschnitt 4.1 vorgestellte Konzept für die Umsetzung eines Engineerings mit unterschiedlichen CAx-Tools und heterogenen EA. Im darauffolgenden Abschnitt wird beispielhaft beschrieben, wie mit Hilfe der Architektur ein versionsbasiertes Modulengineering stattfinden und der Austausch von Engineering-Informationen zwischen Modulhersteller und Systemintegrator bzw. Anlagenbetreiber vereinfacht werden kann.

4.1 Aufbau der Architektur

Die in Bild 3 schematisch dargestellte Virtualisierungsarchitektur umfasst ein Netzwerk an Modellen, Werkzeugen und lauffähigen Lösungen. Den Ausgangspunkt bildet eine Engineering-Lösung (A), welche einem Fertigungsmodul entspricht und im Kontext von Industrie 4.0 als ein Asset aufgefasst wird. Das Modul besitzt eine VWS (B), welche auf einem Server (C) lauffähig ausgeführt wird. Die VWS beinhaltet ein Informationsmodell des Moduls, welches neben allgemeinen Informationen über das Modul auch Informationen über dessen EA (E) und Referenzen zu deren Speicherorten besitzt. Jede Engineering-Lösung besitzt eine virtuelle Repräsentation (F), welche aus der Summe der EA besteht. Einige der EA, wie etwa der SPS-Steuerungscode, finden auch in der realen Engineering-Lösung Anwendung. Die EA selbst sind ebenfalls auf einem Server (G) abgelegt und sind je nach Rechteverwaltung vom Inter-/Intranet aus zugreifbar. Obwohl C und G separat abgebildet sind, kann es sich dabei um einen Server handeln. Es ist jedoch zu beachten, dass die in der VWS repräsentierten Informationen anderen Sicherheitskriterien unterliegen können, wie die EA. Beispielsweise sollte die Artikelnummer eines Moduls ohne Einschränkungen, der Schaltplan oder SPS-Steuerungscode jedoch nur mit passenden Zugriffsrechten zugänglich sein. Die EA eines Moduls werden durch eine Vielzahl proprietärer Datenformate repräsentiert, die eine ebenso große Zahl proprietärer CAx-Tools (H) benötigen, um gelesen und verarbeitet zu werden. Alle notwendigen CAx-Tools für die EA eines Moduls werden innerhalb einer VM (I) gekapselt. Diese kann entweder auf einem lokalen Rechnersystem oder innerhalb einer Cloud-Infrastruktur (J) betrieben werden. Auf diese Weise werden die proprietären EA mit einer Instanz der VM und den notwendigen Berechtigungen zugänglich. Die VM selbst erhält ähnlich wie die Engineering-Lösung ein Informationsmodell (K), auf welches von der VWS der Engineering-Lösung aus zugegriffen werden kann. Um die Engineering-Lösung und die VM bzw. die EA und CAx-Tools miteinander in Verbindung zu setzen, referenzieren sich die entsprechenden Informationsmodelle gegenseitig.

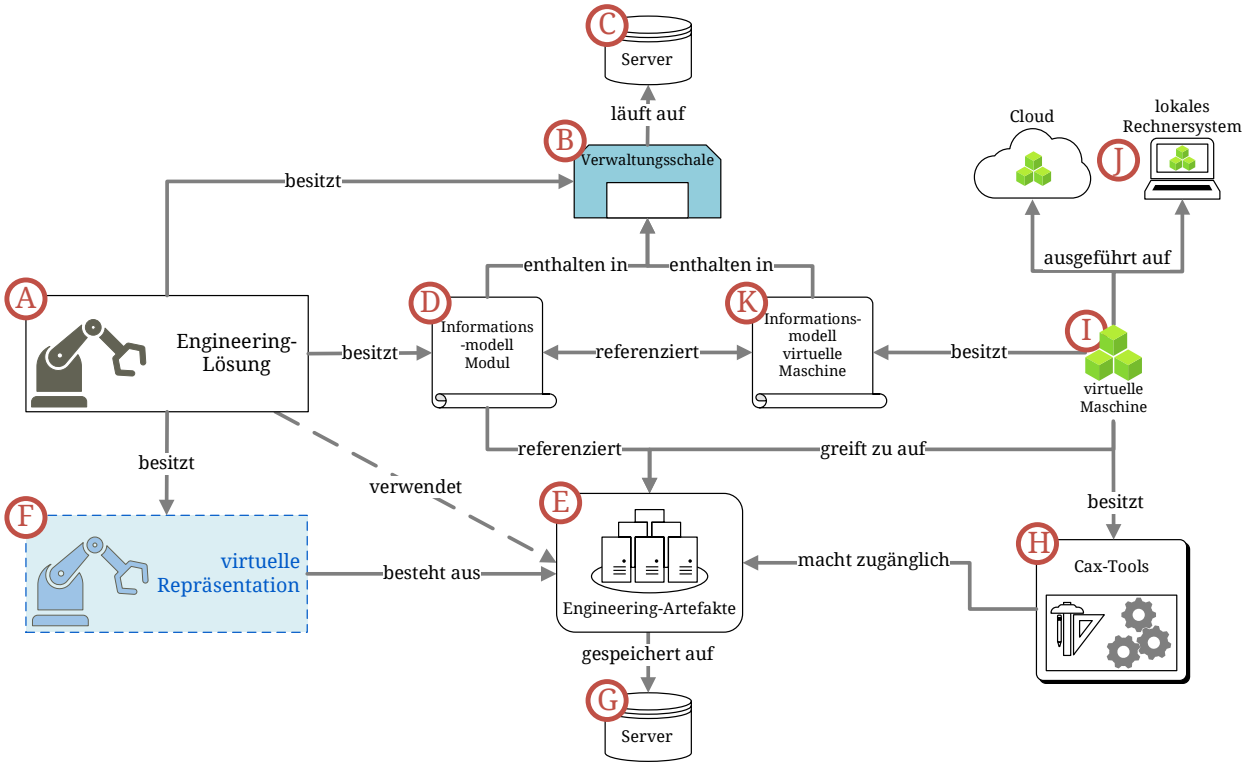

Bild 3 - Darstellung der Virtualisierungsarchitektur

4.2 Anwendungsbeispiel der Architektur

Bei der Anwendung des Konzeptes wird zwischen dem Modul- und Anlagenengineering unterschieden. Während sich das Modulengineering auf die Entwicklung von Modulen fokussiert, also neue EA generiert oder diese weiterentwickelt, werden im Anlagenengineering Module aus einem Pool möglicher Varianten ausgewählt und für den jeweiligen Anwendungsfall konfiguriert. Das betrachtete Szenario ist in Bild 4 schematisch dargestellt.
Im Modulengineerings dient der Server (G aus Bild 3) als Engineering-Artefakt Repository für sämtliche EA des Moduls. Hier sind stets die aktuellen Versionen der verschiedenen EA abgelegt und aus dem Netzwerk zugreifbar. Die EA werden während des Modulengineerings von den Entwicklern des Modulherstellers generiert, weiterentwickelt und mit dem Server (G) synchronisiert (blaue Pfeile in Bild 4). Optimalerweise wird ein echtzeitfähiges paralleles Engineering an den EA ermöglicht und ein lokales Arbeiten vermieden. Beispiel hierfür ist die 3D-CAD-Plattform *OnShape* der Firma PTC[5]. Sollten derartige Mechanismen nicht zur Verfügung stehen, kann auch auf Ein-/Auscheckmechanismen zurückgegriffen werden. Sobald eine Modulversion bereit ist an Kunden/Betreiber vertrieben zu werden, wird eine VM erstellt, welche neben einer Kopie des aktuellen EA-Standes auch alle notwendigen CAx-Tools (in der aktuell verwendeten Version) für die Interaktion mit den EA beinhaltet. Für die Modulversion wird zusätzlich eine Typ-Verwaltungsschale erzeugt.

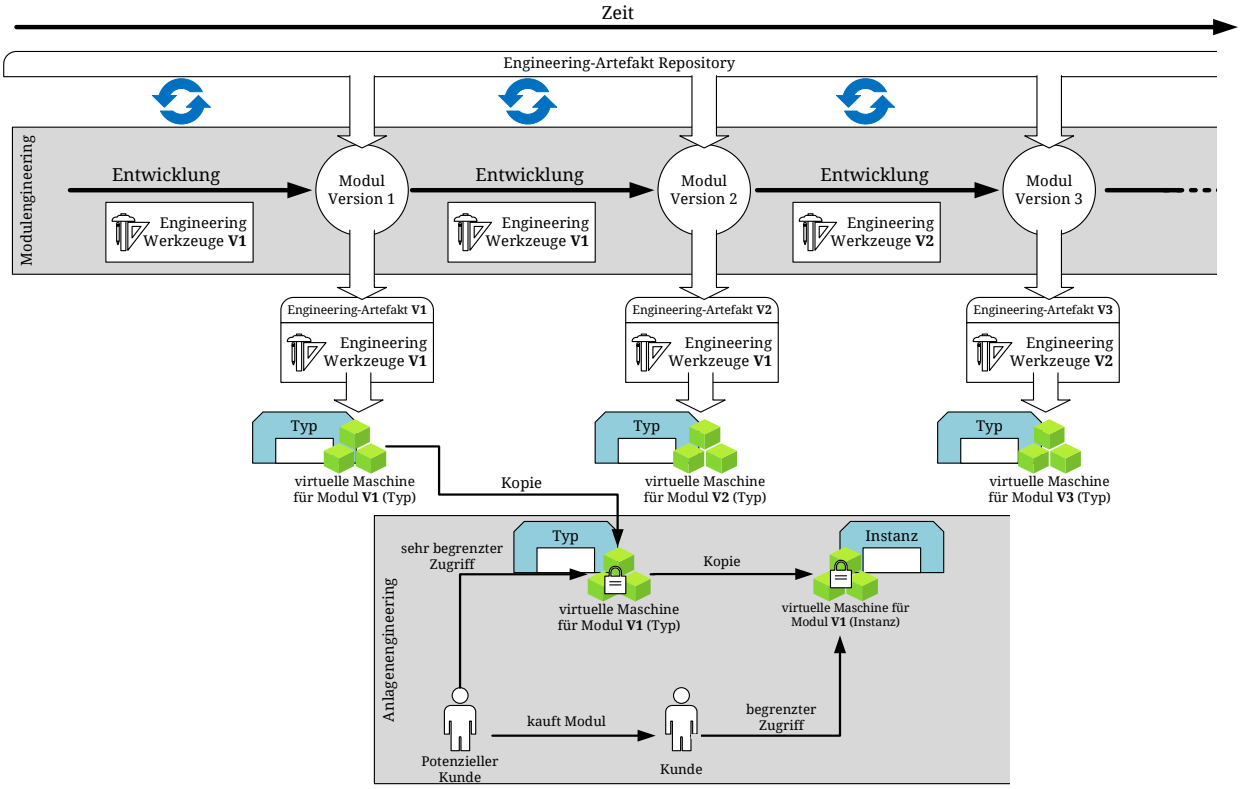

Bild 4 - Schematische Darstellung der Architekturanwendung im Modul-/ Anlagenengineering

Die VM dient als Basis für alle weiteren für diese Modulversion erzeugten Instanzen. Über die entsprechenden Informationsmodelle innerhalb der Verwaltungsschale kann ein potenzieller Kunde eines Moduls generelle Informationen über das Modul und Zugriffsmöglichkeiten auf eine VM des Modultyps erhalten. Durch eine entsprechende Verwaltung der Zugriffsrechte kann der potenzielle Kunde mit Hilfe der Tools den Modultyp in einem ersten Anlauf für seinen Anwendungsfall verifizieren und so die beste Lösung finden. Sobald sich der Kunde für den Kauf des Moduls entscheidet, kann diesem

[5] https://onshape.inneo.com/de/ueberblick.html

eine Kopie der VM mit erhöhten Zugriffsrechten zur Verfügung gestellt werden. So kann der Kunde bereits wenige Minuten nach dem Kauf mit der virtuellen Inbetriebnahme des Moduls beginnen. Aspekte wie Lizenzierungsmechanismen, Knowhow-Schutz und Toolhandhabung werden in Kapitel 6 diskutiert. Parallel zum Vertrieb der Modulversion 1, kann die Entwicklung des Moduls beim Modulhersteller fortgeführt werden. Auf diese Weise entsteht für jede Modulversion eine VM mit zugehörigen EA-/ und CAx-Tool-Versionen. Die Basisversion der VM kann archiviert und bei Bedarf für bspw. Servicedienstleistungen o.ä. als Grundlage für Untersuchungen verwendet werden. Alternativ kann ein Kunde dem Servicemitarbeiter des Modulherstellers Zugang auf die eigene VM gewähren.

5 Prototypische Implementierung

Im Rahmen des METHODS-Projektes entwickeln und erproben die Autoren die vorgestellte Architektur anhand einer prototypischen Implementierung. Als Engineering-Lösung wird dabei ein Grundmodul der CP-Factory (A – bezogen auf Bild 3) von der Firma Festo Didactic SE verwendet. Als Engineering-Artefakt (E) dient ein RobotStudio-Modell des Moduls. Das Modul selbst verfügt über eine VWS (B), die auf der Implementierung der Industrial Digital Twin Association e.V.[6] basiert und als Docker-Container auf einem lokalen NAS (C) ausgeführt wird. Über eine REST-API kann nun auf ein prototypisches Informationsmodell des Moduls (D) zugegriffen werden. Hierin abgelegt ist der Speicherpfad des RobotStudio-Modells und einer virtuellen Maschine (I) mit einer lauffähigen RobotStudio-Installation (H). Beide sind auf dem NAS abgelegt. Ein separates Informationsmodell für die virtuelle Maschine (K) ist im aktuellen Prototyp noch nicht implementiert.

Mit Hilfe des Prototyps können zwei Szenarien der Architektur verifiziert werden.

1. Ein für das RobotStudio-Modell verantwortlicher Modulentwickler findet mit Hilfe des Informationsmodells in der VWS einen Zugriffspunkt für eine VM mit der installierten RobotStudio-Instanz. Sobald sich der Entwickler an der VM anmeldet, wird ein Skript innerhalb der VM gestartet, welches automatisch das RobotStudio-Modell herunterlädt und in einem passenden Verzeichnis abspeichert. Das RobotStudio-Modell entspricht somit der aktuellen Version im Engineering-Artefakt Repository aus Bild 4. Alternativ kann bei Start der VM auch ein automatischer Log-In in eine cloudbasierte CAx-Umgebung stattfinden, welche ein Arbeiten ohne Herunterladen der EA ermöglicht und direkt auf das EA im Repository zugreift. Der Entwickler kann nun seine Arbeit innerhalb der VM fortsetzen. Je nach Variante wird bei Herunterfahren der VM die geänderte Version des RobotStudio Modells in das Engineering-Repository hochgeladen, oder der Entwickler aus der Cloud-Anwendung ausgeloggt.
2. Ein Kunde kann über die VWS generelle Informationen über das Modul erhalten und für eine genauere Untersuchung eine VM-Instanz des Modul-Typs herunterladen. Zugangsdaten für eine Rolle mit beschränkten Rechten sind ebenfalls in dem Informationsmodell zu finden. Alternativ kann statt einem Speicherpfad auch ein Zugriffspunkt im Informationsmodell hinterlegt sein. Auf diese Weise kann sich ein Kunde auf eine vom Modulhersteller bereitgestellte VM einwählen und die EA untersuchen.

[6] https://industrialdigitaltwin.org/

6 Diskussion und Ausblick

Mit Hilfe der vorgestellten Architektur ist es möglich, die Entwicklung von Modulen der Fertigungsindustrie an den Versionscharakter von Modulen anzugleichen. Weiterhin werden Interessengruppen, unabhängig ihrer verwendeten CAx-Toollandschaft, in die Lage versetzt, EA aus dem Modulengineering wiederzuverwenden.
Da es sich bei CAx-Tools meist um komplexe Software handelt, ist seitens der Kunden viel Erfahrung für den sicheren Umgang notwendig. Daher kann es für die o.g. Kunden herausfordernd sein, die unbekannten CAx-Tools zu bedienen. Obwohl mit Hilfe der vorgestellten Architektur zwar nicht mehr alle möglichen CAx-Tools bereitzuhalten sind, ist es dennoch notwendig, geschultes Personal für den Umgang mit den entsprechenden Tools einzusetzen. Allerdings muss generell zwischen dem Entwickeln und dem Konfigurieren/Bewerten einer Engineering-Lösung unterschieden werden. Innerhalb einer Domäne existieren grundlegende Ähnlichkeiten zwischen den verschiedenen CAx-Tools. So sollte es einem Domänenspezialisten möglich sein, grundlegende Aussagen über eine Engineering-Lösung auch in einem fremden CAx-Tool treffen zu können. Neben der Komplexität spielt auch die Lizenzierung der CAx-Tools eine entscheidende Rolle. Durch die Versions- und Kopiermechanismen der Architektur entsteht eine große Zahl von VM mit darauf installierten CAx-Tools, die in vielen Fällen eine kostenpflichtige Lizenzierung benötigen. Besonders statische Lizenzen können hier zu einem wirtschaftlichen Nachteil führen. Daher bietet es sich an, die CAx-Tools mit einem Pay-per-Use Lizenzmodell auszustatten. Während der Modulentwicklungs- und Angebotsphase werden die Lizenzkosten vom Modulhersteller getragen. Sobald ein Modul durch einen Kunden erworben wurde und dieser eine Kopie der entsprechenden VM erhält, gehen auch die Lizenzkosten auf den Kunden über. Die Übergabe von VM mit allen relevanten EA an Kunden kann das Knowhow des Modulherstellers offenlegen. Daher sollen zukünftig entweder umfassende Zugriffsverwaltungen der VM zum Einsatz kommen, oder die EA werden derart verschlüsselt, dass ein Reverse-Engineering nicht möglich ist.
Damit die vorgestellte Architektur Anwendung findet, werden zukünftige Informationsmodelle für Engineering-Lösungen und VM definiert und mit anderen Entwicklungen innerhalb der Plattform Industrie 4.0 abgeglichen, sodass inkompatible Insellösungen vermieden werden. Weiterhin besteht Forschungsbedarf in der Interaktion verschiedener VM, um die Simulation modularer virtueller Anlagen zu ermöglichen. In diesem Zuge müssen auch Mechanismen entwickelt werden, welche die Simulation von Produkten, die auf den Modulen gefertigt werden, in und zwischen den virtuellen Maschinen erlauben. Entsprechend den vorgestellten Projektzielen soll das virtuelle, modulare Engineering anschließend um Mixed-Reality Technologien ergänzt werden, um ein gemischt real-virtuelles Engineering, wie in [14] beschrieben, zu ermöglichen.

Danksagung

Die Arbeiten innerhalb des METHODS-Projektes werden vom Deutschen Bundesministerium für Bildung und Forschung unter dem Förderkennzeichen (13FH511KX9) gefördert. Die Verantwortung für den Inhalt der Arbeit liegt bei den Autoren.

Literatur

[1] H. P. Wiendahl et al., „Changeable Manufacturing - Classification, Design and Operation“, *CIRP Annals*, Jg. 56, Nr. 2, S. 783–809, 2007, doi: 10.1016/j.cirp.2007.10.003.

[2] T. Holm, „Aufwandsbewertung im Engineering modularer Prozessanlagen“. Dissertation, InInstitut für Automatisierungstechnik, Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, Hamburg, 2016. [Online]. Verfügbar unter: https://opus.ub.hsu-hh.de/volltexte/2016/3143/

[3] L. Urbas, S. Bleuel, T. Jäger, S. Schmitz, L. Evertz und T. Nekolla, „Automatisierung von Prozessmodulen: Von Package-Unit-Integration zu modularen Anlagen“, *atp edition*, Jg. 54, 01-02, S. 44–53, 2012, doi: 10.17560/atp.v54i01-02.203.

[4] J. Ovtcharova, „Virtual engineering: principles, methods and applications“ in *Design 2010: 11th International Design Conference*, Dubrovnik, Croatia, 2010, S. 1267–1274.

[5] S. I. Shafiq, C. Sanin, E. Szczerbicki und C. Toro, „Virtual Engineering Object / Virtual Engineering Process: A specialized form of Cyber Physical System for Industrie 4.0“, *Procedia Computer Science*, Jg. 60, S. 1146–1155, 2015, doi: 10.1016/j.procs.2015.08.166.

[6] Hochschule Pforzheim, *Aktuelle Projekte am Institut für Smart Systems and Services (IOS3)*. [Online]. Verfügbar unter: https://www.hs-pforzheim.de/forschung/institute/institut_fuer_smart_systems_und_services_ios3/aktuelle_projekte (Zugriff am: 3. Dezember 2021).

[7] P. Mandl, *Grundkurs Betriebssysteme: Architekturen, Betriebsmittelverwaltung, Synchronisation, Prozesskommunikation, Virtualisierung*, 4. Aufl. Wiesbaden: Springer Vieweg, 2014. [Online]. Verfügbar unter: http://swbplus.bsz-bw.de/bsz41643441xcov.htm

[8] Plattform Industrie 4.0, „Details Of the Administration Shell: Part 1 - The exchange of information between partners in the value chain of Industrie 4.0“, Plattform Industrie 4.0, Berlin, Nov. 2020. [Online]. Verfügbar unter: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html. Zugriff am: 4. Oktober 2021.

[9] Plattform Industrie 4.0, „Details of the Asset Administration Shell: Part 2 - Interoperability at Runtime – Exchanging Information via Application Programming Interfaces“, Plattform Industrie 4.0, Berlin, Nov. 2021. [Online]. Verfügbar unter: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part_2_V1.html. Zugriff am: 25. November 2021.

[10] H. P. Breivold, A. Jansen, K. Sandstrom und I. Crnkovic, „Virtualize for Architecture Sustainability in Industrial Automation“ in *2013 IEEE Computational Science and Engineering (CSE)*, Sydney, Australia, 2013, S. 409–415, doi: 10.1109/CSE.2013.69.

[11] M. Azarmipour, J. Grothoff, H. Elfahaam, U. Epple und C. Gries, „Hypervisor-basierte Virtualisierung in der industriellen Automation“ in *Automation 2018*, Baden-Baden, 2018, S. 467–480, doi: 10.51202/9783181023303-467.

[12] M. Azarmipour, H. Elfaham, J. Grothoff, C. von Trotha, C. Gries und U. Epple, „Dynamic Resource Management for Virtualization in *Industrial Automation“ in 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, D.C., DC, USA, 2018, S. 2878–2883, doi: 10.1109/IECON.2018.8591622.

[13] H. P. Breivold und K. Sandstrom, „Virtualize for test environment in industrial automation“ in *2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Barcelona, Spain, 2014, S. 1–8, doi: 10.1109/ETFA.2014.7005089.

[14] G. Hildebrandt, P. Habiger und R. Drath, „Virtual-In-The-Loop-Engineering: A categorisation and terminology for modular plants and interfaces“ in *2021 IEEE 26th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vasteras, Sweden, 2021, S. 1–4, doi:10.1109/etfa45728.2021.9613589.