

Generierung digitaler Modelle zur virtuellen Inbetriebnahme und Qualifizierung der Steuerungssoftware verfahrenstechnischer Anlagen

Markus Vogler, Jan Nagel und Benjamin Massow

Zusammenfassung—Anlagen in der verfahrenstechnischen Industrie werden laufend an die sich ändernden Anforderungen angepasst. Besonders in der pharmazeutischen Industrie ist die Prozessoptimierung unverzichtbar. Aufgrund der sehr hohen Qualitätsansprüche an Arzneimittel muss gewährleistet sein, dass die genutzte Software den regulatorischen Vorgaben entspricht. Dies erfordert ressourcenintensive Software-Validierungsprozesse. Die begrenzten Stillstandszeiten erlauben nur kurze Testphasen an der realen Anlage. Eine vorzeitige Inbetriebnahme der Software an einem virtuellen Modell ist daher notwendig. Die manuelle Erstellung dieser Modelle benötigt Fachwissen und Zeit. Um Engpässe zu vermeiden, wird in dieser Arbeit ein automatisierter Ablauf zur Modellerstellung erarbeitet. In der Konzeptentwicklungsphase umfasst die Aufgabe nicht nur das Design und ein parametrisches Modellierungsprogramm, sondern auch die Wahl eines standardisierten Dateiformats. Nach der Modellerstellung wird das virtuelle Modell mit der gewählten Steuerungssoftware verbunden. Im letzten Schritt, der Softwarequalifizierung, wird die implementierte Ablaufsteuerung gegen die Funktionsspezifikation getestet. Die Umsetzung wird anhand einer Fallstudie gezeigt.

Schlagwörter—automatische Modellerstellung, virtuelle Inbetriebnahme, R&I-Schemen, Softwarequalifizierung.

I. EINLEITUNG

UM den permanent steigenden Anforderungen gerecht zu werden, sind moderne verfahrenstechnische Anlagen mit größtmöglicher Flexibilität

M. Vogler studiert am Studiengang Mechatronik, MCI, Innsbruck, Österreich, e-mai: m.vogler@mci4me.at

zu planen. Dennoch sind im Verlauf des Anlagenlebenszyklus unvermeidbare Anpassungen an Software und Hardware („Changes“) erforderlich. Diese können ausschließlich während Anlagenstillständen durchgeführt werden.

Für Automatisierungstechniker/innen bleibt so nur ein kleines Zeitfenster zum Testen und Optimieren der Ablaufsteuerung an der realen Anlage, bevor die Produktion wieder aufgenommen wird. Aufgrund dieser Gegebenheit müssen alle fundamentalen Abläufe zum Zeitpunkt der Tests am realen System programmtechnisch umgesetzt, simuliert und qualifiziert sein. Zur Durchführung dieser zeitintensiven Tätigkeiten wird das Konzept der virtuellen Inbetriebnahme herangezogen. Dabei wird der gesamte Produktionsprozess in einer virtuellen Umgebung nachgebildet und mit der realen Steuerung verbunden. Die Simulation der Prozesse in Echtzeit ermöglicht es den Technikern/innen, Szenarien durchzuspielen und potenzielle Probleme frühzeitig zu erkennen. Eine der größten Herausforderungen im Zuge der virtuellen Inbetriebnahme ist die Erstellung einer geeigneten Simulationsumgebung, welche oft manuell durchgeführt wird. Mit dem Einsatz digitaler Ingenieursmethoden soll im Folgenden ein Workflow zur automatischen Modellgenerierung (AMG) erarbeitet werden.

Der Einsatz der AMG steigert die Effizienz der virtuellen Inbetriebnahme und entlastet Ingenieure/innen. Die Vorteile der Ressourceneinsparung gel-

ten nicht nur für die Erstellung des Simulationsmodells, sondern auch für laufende Änderungen. Ein modulares Baukastensystem ermöglicht auch weniger erfahrenen Personen die Erstellung hochwertiger Simulationsmodelle.

In dieser Arbeit wird eine Methode zur AMG für die virtuelle Inbetriebnahme in der Prozessindustrie entwickelt. Der Fokus liegt auf der Modellerstellung zur Qualifizierung der Ablaufsteuerungssoftware. Der AMG-Workflow soll modular und erweiterbar sein und fundamentale Designdokumente aus der verfahrenstechnischen Industrie nutzen.

II. STAND DER TECHNIK

In diesem Kapitel werden Methoden zur Erzeugung eines Verhaltensmodells zur virtuellen Inbetriebnahme präsentiert. Diese beschränken sich auf die Prozessindustrie, da die vorgestellten Arbeitsabläufe auf prozessindustriespezifischen Designdokumenten wie R&I-Schemen, Rohrleitungsplänen und 3D-Anlagenmodellen basieren.

A. Designdokumente

Informationen zum verwendeten Equipment, den Rohrleitungen und den geometrischen Abmessungen können aus dem 3D-Modell einer Anlage ermittelt und für die automatische Generierung eines Simulationsmodelles verwendet werden [1], siehe Abbildung 1.

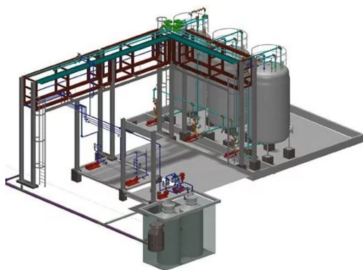


Abbildung 1. Beispielhaftes 3D-Modell einer Anlage aus dem verfahrenstechnischen Umfeld [2].

Eine weitere Möglichkeit ist die Verwendung von R&I-Schemen als Designdokumente [3], [4]. In R&I-Schemen werden alle relevanten Komponenten, Rohrleitungen und Sensoren abgebildet, siehe Abbildung 2.

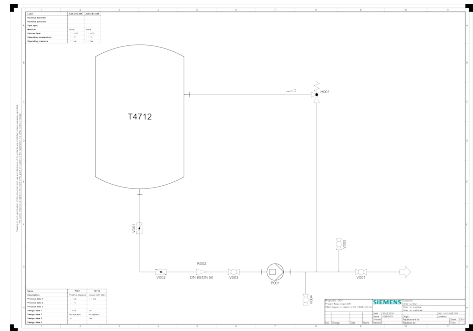


Abbildung 2. Beispielhaftes R&I-Schema zur Beschreibung eines Tank-Pumpen-Kreislaufes [5].

B. Interpretation der Designdokumente

Das Umwandeln des Designdokuments in eine maschinenlesbare Datei ist der nächste Schritt für die Weiterverwendung der Daten. Dazu muss im Vorhinein ein geeignetes Dateiformat ausgewählt werden. Im Folgenden sind einige Anforderungen an das Format aufgelistet:

- hoher Informationsgehalt
- konsistente Datenstruktur
- Skalierbarkeit
- objektorientierter Aufbau
- herstellerunabhängiges Format

Zum Beispiel können *Comma Separated Values* (CSV) oder *Java Script Object Notations* (JSON) direkt aus der CAD-Software exportiert werden. Eine Alternative ist die Speicherung der Informationen im *extensible Markup Language*-Format (XML). Dieses kann als erweiterbares Format zur Speicherung von hierarchisch strukturierten Daten verstanden werden. Eine einheitliche Struktur erreicht man durch die Verwendung von *XML Schema Definitionen* (XSD), deren Regeln eingehalten werden müssen. Beschränkt

auf 3D-Designdokumente ist die Verwendung des *Standard for the Exchange of Product Data* - Formats (STEP) oder *Piping Component Files* (PCF).

C. Interpretationsmethoden

Einige Dateiformate können direkt aus Planungsprogrammen in Form einer maschinenlesbaren Datei exportiert werden. Für 3D-basierte Designdokumente wird in diesem Zusammenhang oft das universelle STEP-Format verwendet.

Da speziell bei Bestandsanlagen die technische Dokumentation oft nur in Form von R&I-Schemen im PDF-Format vorliegt, werden im folgenden Kapitel alternative Interpretationsmethoden für R&I-Schemen angeführt.

Liegt das R&I-Schema als Rastergrafik vor, muss vorerst eine Vektorisierung durchgeführt werden. Dazu ist der Einsatz einer leistungsstarken Bildverarbeitungssoftware notwendig [6]. Dabei wird die Bildverarbeitungssoftware mit Symbolvorlagen gefüttert und die Rastergrafik unter Anwendung von künstlicher Intelligenz und „machine learning“ ausgelesen.

Es können auch vektorbasierte R&I-Schemen in XML-Dateien umgewandelt werden [3]. Die Funktionsweise beruht ebenfalls auf dem Prinzip des *template matching* und der *sliding-window*-Methode [7], wobei diese aufgrund der Gegebenheit, dass es sich um Vektorgrafiken handelt, deutlich zuverlässiger und weniger anfällig für Fehlinterpretationen ist. Das Resultat sämtlicher Methoden ist entweder eine maschinenlesbare Datei, siehe Abschnitt II-C, oder eine Objektstruktur, die mühelos verarbeitet werden kann.

D. Modellgenerierung und Visualisierung

In diesem Abschnitt werden Methoden zur automatischen Erstellung des virtuellen Modells aus der maschinenlesbaren Datei, die aus der Interpretation der Designdokumente hervorgeht, vorgestellt. Beispielsweise kann ein Simulationsmodell erstellt werden, indem zu jedem Eintrag einer XML-Datei eine Instanz in einer Simulationssoftware generiert

wird. Die Parameter für die Simulationskomponente werden dabei der XML-Struktur entnommen. Für die XML-Struktur können Schemendefinitionen wie ProteusXML, CAEX oder AutomationML herangezogen werden. Simulationsprogramme sind Simic oder Modelica [8]. Zur Visualisierung wird die zweidimensionale Benutzeroberfläche der jeweiligen Simulationssoftware herangezogen. Eine weitere Möglichkeit der Visualisierung ist die Verwendung von dreidimensionalen Modellen oder der Einsatz von virtual - oder augmented Reality.

E. Verbindung zum Steuerungssystem

Nachdem das Simulationsmodell aus Komponentenbibliotheken zusammengestellt worden ist, muss es mit einem Steuerungssystem verbunden werden. Im Zuge der virtuellen Inbetriebnahme wird zwischen *Soft- bzw. Hardware in the Loop* - Simulationen (SIL/HIL) unterschieden. SIL-Simulationen beschränken sich auf rein virtuelle Abläufe und Komponenten. Im Gegensatz dazu wird bei HIL-Simulationen mindestens ein Teil des Systems mit physischer Hardware realisiert.

Unabhängig von der Art der Simulation müssen die Ausgänge der virtuellen Komponenten aus der parametrischen Modellerstellung auf die Ein- und Ausgänge einer Logik gemappt werden. Die Verbindung zum Steuerungssystem kann, abhängig von der verfügbaren Dokumentation, manuell oder automatisch passieren.

III. KONZEPTERSTELLUNG

Im folgenden Kapitel wird ein möglicher Weg zur automatischen Generierung eines Verhaltensmodells basierend auf einem R&I-Schema gezeigt. Es sollen bisher manuell durchgeführte Schritte automatisiert und Schnittstellen zwischen den einzelnen Prozessen geschaffen werden. Das R&I-Schema wird in erster Linie von externen Quellen bezogen. Dabei ist wichtig sicherzustellen, dass das erhaltene Schema aktuell ist und den qualitativen Anforderungen entspricht. Für das objektorientierte Dateiformat wird

auf die *Data Exchange in the Process Industry-Initiative* (DEXPI) zurückgegriffen. Diese basiert auf dem internationalen Standard für die Integration von Informationen über Anlagen und Anlagenlebenszyklen, ISO 15926. Es handelt sich dabei um eine ontologische Datenmodellierungssprache.

A. Definition von Symbolvorlagen

Für die Umwandlung eines R&I-Schemas in eine DEXPI konforme XML-Datei muss vorerst eine Konfiguration aus Symbolen angelegt oder importiert werden. Dieser Vorgang wird mithilfe der ModelBroker-Software von *Semantum* erstellt, indem Linienzüge angewählt und als Template gespeichert werden, siehe Abbildung 3. Der wichtigste Parameter ist in diesem Arbeitsschritt das *Target*. Dieser legt fest, als welche Klasse das Symbol in der finalen DEXPI konformen, ProteusXML-Datei abgelegt wird.

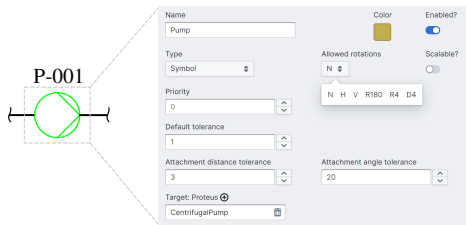


Abbildung 3. Definition einer Vorlage für ein Pumpensymbol (links). Eingabemaske zur Parametrisierung (rechts)

Im Zuge der Templatedefinition müssen auch die Anschlüsse der Komponenten definiert werden. Diese sind im weiteren Verlauf, speziell beim Verbinden der Komponenten essentiell.

B. Symbolvorlagendefinition für Rohrleitungen

Nachdem alle relevanten Symbole erkannt worden sind, wird im nächsten Schritt nach den Verbindungselementen gesucht. Wie bei der Definition von Symbolvorlagen wird ein Abschnitt der Rohr- oder Informationsleitung als Symbolvorlage definiert.

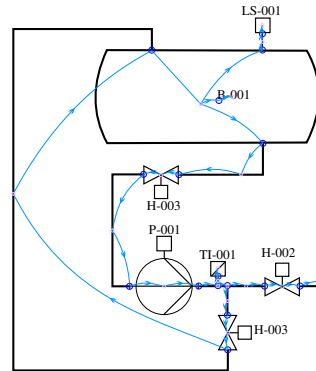


Abbildung 4. Interpretiertes R&I-Schema mit erkannten Verbindungen. Diese werden mithilfe der blauen Pfeile dargestellt.

Im nächsten Schritt werden die Komponenten durch Ausführen des *Target* Befehls miteinander verbunden. Das Ergebnis wird visuell durch blaue Pfeile dargestellt, siehe Abbildung 4. Die Pfeilrichtung stellt dabei gleichzeitig die Flussrichtung des Mediums dar. Danach kann die ProteusXML-Datei exportiert werden.

C. XML-Deserialisierung

Nachdem die Proteus XML-Datei erstellt, exportiert und verifiziert worden ist, muss sie zur weiteren Verarbeitung in eine objektorientierte Struktur umgewandelt werden. Dieser Vorgang wird Deserialisierung genannt und wird mithilfe der Programmierumgebung VisualStudio unter Verwendung der Programmiersprache C# realisiert.

D. Equipmentgenerierung

Zur Visualisierung und Erstellung des virtuellen Modells wird die Entwicklungsumgebung Unity mit zugehörigen Bibliotheken verwendet. Für die Erstellung des Equipments iteriert eine Schleife über die Einträge des deserialisierten XML-Objekts. Alle relevanten Eigenschaften werden in einem temporären Objekt abgespeichert, mit denen eine Funktion zur

Generierung des Equipments parametrisiert wird. In dieser Funktion wird auf Basis der übergebenen Daten eine Bibliothekskomponente an der hinterlegten Position geladen, nach der *TagNameAssignmentClass* benannt und mit den übergebenen Werten initialisiert. Ein Beispiel dafür ist in Abbildung 5 ersichtlich. Dabei sind ein Tank und eine Pumpe generiert worden.

E. Etablieren der Verbindungen

Die Verbindung der Komponenten wird in der XML-Datei in den Objekten vom Typ *PipingNetworkSegments* festgehalten. Über diese wird mit einer inneren Schleife iteriert und die Informationen zur Position (x,y,z), zur Verbindung (*FromID*, *ToID*), zur Rohrleitung (*CenterLine*) und zu den involvierten Komponenten (*PipingComponents*) werden extrahiert.

Zur Erstellung eines Flowsheets werden vorerst alle *PipingComponent-IDs* aufgelistet, wobei sich am Anfang die *FromID* und am Ende die *ToID* finden. Als Nächstes wird zwischen den Einträgen ein Objekt vom Typ *CenterLine_n* eingefügt. Diesem wird im selben Schritt eine *IN_ID* und *OUT_ID* angehängt und mit der vorherigen bzw. nachfolgenden ID verbunden.

Im nächsten Schritt wird über die eben erstellte Anschluss-ID Liste iteriert und für jedes Objekt vom Typ *CenterLine* eine entsprechende Rohrgeometrie an der *Position* geladen. Zeitgleich wird mit einer Schleife über den gesamten Strukturbaum iteriert und nach Objekten mit gleichen Namen wie die *IN.ID/ OUT.ID* des *CenterLine*-Elements gesucht. Es erfolgt ein gegenseitiges Belegen der Anschlüsse des Suchobjektes mit den Anschlüssen der Rohrleitung, siehe Abbildung 5.

F. Instrumentengenerierung

Ähnlich wie bei der Erstellung der Rohrleitungen werden alle Informationen zu den Sensoren und den Signalverbindungen aus der Objektstruktur extrahiert. Dann werden ebenfalls Verhaltensmodelle für Sensoren in der Unity Szene geladen. Es werden

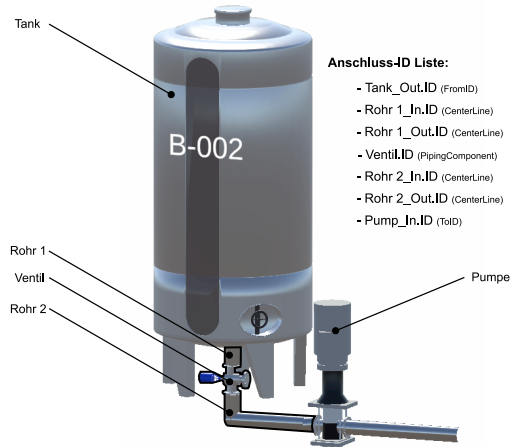


Abbildung 5. Prinzip eines *PipingNetworkSystems* bzw. eines *PipingNetworkSegments* anhand eines Minimalbeispiels. Dieses besteht aus zwei Rohrleitungsstücken und einem Ventil und verbindet Equipment miteinander.

Sensoren zur Messung der physikalischen Größen Temperatur und Füllstand unterstützt.

G. Verbindung zur Automatisierungssoftware

Die ADS-Schnittstelle zwischen TwinCAT und Unity ermöglicht eine nahtlose Verbindung zwischen der Ablaufsteuerung und der Unity-Umgebung. Nach der Erstellung eines lauffähigen Simulationsmodells in Unity, kann mithilfe des *Projektgenerator*-Modules des PlanX-Assistants ein TwinCAT Projekt erstellt werden. Die Verbindungen der Verhaltensmodelle müssen in TwinCAT noch über die Kommunikationsschnittstelle zugeordnet werden. Dies ist nur möglich, wenn ein BUS in TwinCAT konfiguriert ist. Bei Verwendung von Profinet kann dieser BUS mithilfe von PlanX automatisch erstellt werden. Um eine reibungslose Kommunikation zu gewährleisten, müssen die Bus-E/As auf die entsprechenden Ein- und Ausgänge der aktiven Verhaltensmodellobjekte

wie Ventile und Sensoren abgebildet werden. Im Zuge dieser Arbeit wird die SPS in Abbildung 6 mithilfe eines zweiten TwinCAT Projektes umgesetzt. In diesem Projekt wird neben der Ablaufsteuerung, siehe Kapitel IV, auch die Visualisierung in Form einer graphischen Benutzeroberfläche implementiert. Zur Umsetzung der gewünschten Funktion sind also ein Unity Projekt und zwei TwinCAT Projekte notwendig. Die Kommunikation zwischen Unity und TwinCAT wird mithilfe des PlanX-Assistants realisiert.

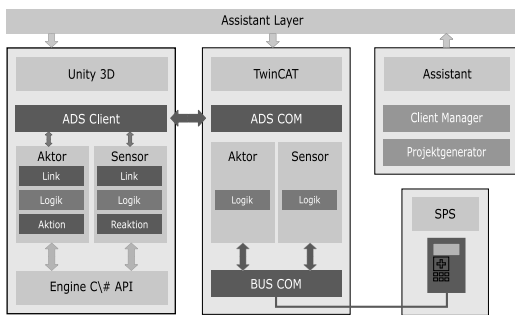


Abbildung 6. Funktionsweise des PlanX-Assistants. Der Assistant Layer ermöglicht die Kommunikation und die Generierung von für TwinCAT-Projekten [9].

IV. ENTWICKLUNG EINER ABLAUFSTEUERUNG

Ablaufsteuerungen in der verfahrenstechnischen Industrie werden meist in Form von Schrittketten realisiert. Eine Schrittkette ist eine sequentielle Abarbeitung verschiedener Zustände eines Systems. Prinzipiell handelt es sich bei einer Schrittkette um eine alternative Darstellungsmöglichkeit eines Zustandsautomaten (State-Machine).

A. Erstellung von Designdokumenten für die Ablaufsteuerung

Im Lastenheft werden die Anforderungen des Auftraggebers an den Auftragnehmer festgehalten. Im verfahrenstechnischen Umfeld wird das Lastenheft daher oft vom Betreiber der Anlage angefertigt und

an externe Dienstleister ausgehändigt. Bei einem Lastenheft handelt es sich meist um ein textbasiertes Dokument, welches mit Bildausschnitten aus dem verfahrenstechnischen Schema visuell unterstützt wird. Daraufhin wird vom Dienstleister ein Pflichtenheft erstellt. Im Rahmen der Funktionsplanung werden im Pflichtenheft oder der *Function Specification* (FS) alle Anforderungen festgehalten und zur Umsetzung vorbereitet. Im Allgemeinen ist das Pflichtenheft für die Planung von Schrittketten oder Ablaufsteuerungen ein tabellarisch aufgebautes Dokument. Eine mögliche Art der Umsetzung zeigt Tabelle I.

Schritt	V1	V2	V3	Trans.
S0	0	0	0	V1 & V2 & V3 = ZU
S1	1	-	-	V1 = AUF
S2	0	1	-	V1 = ZU & V2 = AUF
S3	-	0	1	V2 = ZU & V3 = AUF
S4	-	-	0	V3 = ZU

Tabelle I
EINDEUTIGE DEFINITION DES ABLAUFES IN TABELLENFORM. V1-3 STELLEN DABEI AKTOREN DAR.

B. Implementierung der Schrittkette

Die Implementierung der Schrittkette wird direkt in der TwinCAT- Umgebung vorgenommen. Der Ablauf ist durch das Pflichtenheft aus dem vorherigen Abschnitt festgelegt. In TwinCAT wird dies durch Hinzufügen einer Programm-Organisationseinheit vom Typ *SFC Sequential Function Chart* (SFC) realisiert. Dies ermöglicht eine grafikunterstützte Programmierung, im Zuge derer einzelne Befehlsblöcke verbunden werden müssen.

C. Prüfvorschrift

Die Prüfvorschrift dient als Grundlage für die Durchführung des Konfigurations-, Funktions- und Anforderungstests. Bei der Prüfvorschrift handelt es sich um ein papierbasiertes Dokument, welches vom Auftraggeber erstellt und dem Auftragnehmer zur

Überprüfung bereitgestellt wird. Im Zuge des Tests wird geprüft, ob die technischen Spezifikationen aus Abschnitt IV-A eingehalten werden.

V. FALLSTUDIE

Im Folgenden wird die Ausarbeitung eines Musterbeispiels dokumentiert. Das Beispiel bezieht sich auf ein Mischwerk, bei dem ein warmes und ein kaltes Medium kombiniert werden, um ein Endprodukt mit vorgegebener Temperatur zu erzeugen. Das zugehörige R&I-Schema ist in Abbildung 7 ersichtlich. Es stehen drei Tanks zur Verfügung, wobei die zwei Medientanks (B-001 und B-002) ein Fassungsvermögen von jeweils 50 Litern und der Mischtank (B-003) ein Fassungsvermögen von 150 Litern besitzt. Der Medientransfer zwischen den Tanks wird mithilfe einer Zentrifugalpumpe (P-001) realisiert. Um die Prozessgrößen Füllstand und Temperatur zu bestimmen, befinden sich an den Tanks jeweils ein Füllstandssensor (LI-00x) und ein Temperatursensor (TI-00x).

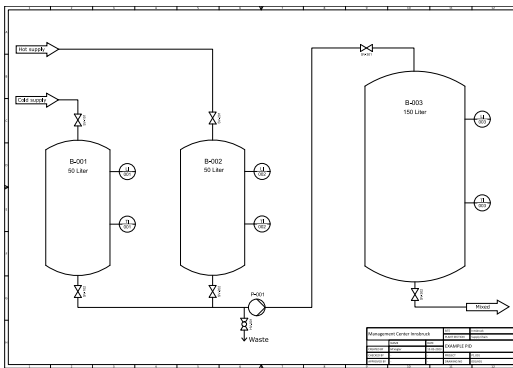


Abbildung 7. R&I-Schema zur Fallstudie

Im ersten Schritt der Umsetzung muss das R&I-Schema mithilfe der ModelBroker-Software und unter Anwendung der Schritte aus Abschnitt III in ein XML-Dokument umgewandelt werden. Nach Anlegen einer Konfiguration, bestehend aus allen vorkommenden Symbolen, kann ein *template matching*

durchgeführt werden. Für das Etablieren der Verbindungen wird nach den Schritten aus Abschnitt III-B vorgegangen.

Für die parametrische Modellerstellung mit Unity muss der Dateipfad des XML-Dokuments mit zugehöriger XSD-Datei vorgegeben werden. Im Hintergrund werden die entsprechenden Verhaltensmodelle aus der Unity Bibliothek geladen, parametrisiert und mit den Algorithmen aus Abschnitt III-G verbunden, siehe Abbildung 8.

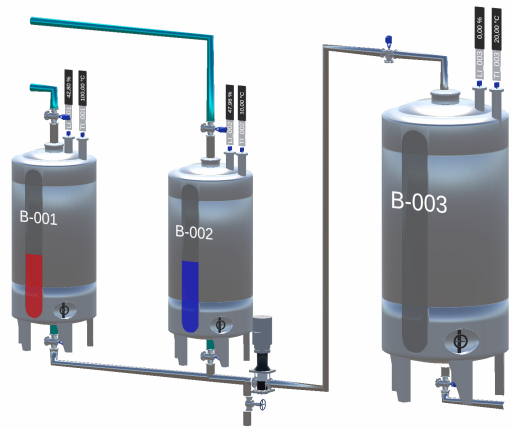


Abbildung 8. Parametrisch generiertes Simulationsmodell der Anlage, welche durch das R&I-Schema von Abbildung 7 charakterisiert wird.

Im nächsten Schritt muss das erstellte Modell mit der Steuerungssoftware verbunden werden. Dafür stehen TwinCAT und der PlanX-Assistent aus Kapitel III-G zur Verfügung. Unter Verwendung des PlanX-Assistent wird nun ein TwinCat-Projekt angelegt. Die Funktion kann durch manuelles Ändern der Parameterwerte in TwinCAT oder Unity überprüft werden. Bei korrekter Ausführung der erwähnten Schritte werden daraufhin Aktoren in der Unity-Szene betätigt bzw. Sensorwerte in TwinCAT aktualisiert.

Um den gewünschten Ablauf zu erhalten, wird eine Kopie der Pflichtenheftvorlage aus Kapitel IV-A mit den notwendigen Informationen befüllt. Nachdem

das Simulationsmodell mit TwinCat verbunden und die Schrittkette implementiert worden sind, kann der Ablauf simuliert werden.

Parallel dazu werden die einzelnen Prüfpunkte auf der Prüfvorschrift aus Kapitel IV-C bzw. die spezifizierten Schritte des Pflichtenhefts aus Kapitel IV-A abgehakt. Können alle Testpunkte positiv abgeschlossen werden, gilt die Software für die Ablaufsteuerung als qualifiziert und kann dem Auftraggeber übergeben werden.

VI. ZUSAMMENFASSUNG UND AUSBLICK

Durch Anwendung der erarbeiteten Methoden aus Kapitel III kann erfolgreich ein Modell zur Qualifizierung der Ablaufsteuerung entwickelt werden. Der Workflow besteht dabei aus einer Reihe sequentieller Arbeitsschritte, die ohne Adaptionen auf unterschiedliche R&I-Schemen angewendet werden können. Es können folgende Erkenntnisse gewonnen werden:

- Die Interpretation der Designdokumente kann durch die Verwendung von geeigneten Programmen zu einem hohen Grad automatisiert werden.
- Die parametrische Modellerstellung kann unter Voraussetzung von existierenden Bibliothekskomponenten weitgehend autonom durchgeführt werden.
- Mithilfe der verfügbaren Werkzeuge und der eigenen C#-Schnittstellen kann eine deutliche Reduktion des Zeitbedarfes erreicht werden.
- Durch die AMG werden zufällige Fehler durch menschliche Eingriffe ausgeschlossen.

Die zukünftigen Arbeitsschritte umfassen die Erweiterung der C#-Schnittstelle, der Unity-Komponenten-Bibliothek und der Design- sowie Prüfdokumentation. Außerdem werden alle Teilschritte durch die Verwendung komplexerer R&I-Schemen getestet und gegebenenfalls angepasst, um die Gesamtsystem-Robustheit und -Zuverlässigkeit sicherzustellen. Im Hinblick auf die zukünftige Verwendung des virtuellen Modells als Digitaler Zwilling eröffnen sich zusätzliche Möglichkeiten. Das virtuelle Modell könnte über den gesamten

Lebenszyklus hinweg genutzt werden, von der Konzeption und Entwicklung bis hin zu Schulungs- oder Optimierungszwecken.

LITERATUR

- [1] G. Santillan, S. Sierla, T. 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*, 2018.
- [2] H.-J. Wächter and G. Schmidt. Rohrleitungsplanung. [Online]. Available: <https://prozesstechnik.industrie.de>
- [3] M. Azangoo, L. Sorsamaki, S. A. Sierla, T. Matasniemi, M. Rantala, K. Rainio, and V. Vyatkin, "A methodology for generating a digital twin for process industry: A case study of a fiber processing pilot plant," *IEEE Access*, 2022.
- [4] H. Mayer, "Enable building system simulation by deriving a Digital Twin for operation from 2D P&ID plans," in *Proceedings of Building Simulation 2021: 17th Conference of IBPSA*, 2021.
- [5] DEXPI - Data Exchange in the Process Industry. Dexpi - code, data and specification repository. [Online]. Available: <https://gitlab.com/dexpi/TrainingTestCases>
- [6] F. Stinnera, M. Wieceka, M. Baranskia, A. Kümpel, and D. Müller, "Automatic digital twin data model generation of building energy systems from piping and instrumentation diagrams," 2021.
- [7] S. Kang, E. Lee, and H. Baek, "A Digitization and Conversion Tool for Imaged Drawings to Intelligent Piping and Instrumentation Diagrams (P&ID)," *Energies*, 2019.
- [8] M. Barth and A. Fay, "Automated generation of simulation models for control code tests," *Control Engineering Practice*, 2013.
- [9] P. Mundhenke, "Konzeptionierung einer synchron nutzbaren WebGL-Anwendung für die Präsentation eines Anlagenmodells," *Hochschule Hannover - University of Applied Sciences and Arts*, 2023.



Markus Vogler besucht den Masterstudiengang für Mechatronik am MCI Innsbruck/Österreich. Beruflich beschäftigt er sich mit der Planung und Konfiguration von Prozessleitsystemen in der pharmazeutischen Industrie.