

Simulation in Produktion und Logistik 2023
Bergmann, Feldkamp, Souren und Straßburger (Hrsg.)
Universitätsverlag Ilmenau, Ilmenau 2023
DOI (Tagungsband): 10.22032/dbt.57476

Integration eines SAP-Transportleitsystems in eine Simulationsumgebung zur Unterstützung von Entscheidungen im operativen Betrieb

Integration of an SAP Transport Control System in a Simulation Environment to Support Decisions in Operational Management

Stefan Galka, Florian Schmid, OTH Regensburg, Regensburg (Germany),
stefan.galka@oth-regensburg.de, florian.schmid@oth-regensburg.de

Dominik Grasser, Jonathan Hohm, Flexus AG, Würzburg (Germany),
d.grasser@flexus.net, j.hohm@flexus.net

Stephan Stauber, SimPlan AG, Regensburg (Germany), stephan.stauber@simplan.de

Sebastian Meißner, Fachhochschule Landshut, Landshut (Germany),
sebastian.meissner@haw-landshut.de

Abstract: Digital twins enable the digital representation of physical systems and provide comprehensive modelling and analysis capabilities. An online data exchange between the Digital twin and SAP is feasible, through integration of a transport control systems (SAP) via RESTful API. This interface allows decision making in resource allocation and furthermore the adaptation of the digital model if the real system changes. Despite the associated challenges in terms of data synchronisation, the integration has benefits such as a reduction in modelling effort and an increase in the lifetime of the digital twin. In the context of this work, the design of the interface between the simulation software and a SAP transport control system via RESTful API is described.

1 Simulationsbasierte Digitale Zwillinge für die Unterstützung von operativen Entscheidungen

Interne Transportsysteme sind ein wesentlicher Bestandteil von intralogistischen Systemen. Aufgrund der flexiblen Einsatzmöglichkeiten werden häufig Fahrzeugsysteme aus Flotten von Gabelstaplern, Routenzügen und Fahrerlosen Transportfahrzeugen eingesetzt. Diese Systeme sind durch einen hohen Grad an Komplexität gekennzeichnet, da es eine Vielzahl von Wechselwirkungen mit den vor- und nachgelagerten Systemen gibt und die Anforderungen sich schnell verändern. (Fottner et al. 2022)

Somit können die Auswirkungen von Entscheidungen, wie z.B. die eingesetzte Fahrzeuganzahl, von den Verantwortlichen nur schwer eingeschätzt werden. Simulationsbasierte Digitale Zwillinge können die Entscheidungsträger unterstützen, in dem Entscheidungen vorab mit Hilfe des Digitalen Zwillings untersucht werden. (Kauke et al. 2021) In diesem Beitrag beschreibt der Begriff Digitaler Zwilling ein virtuelles Abbild eines realen Systems, welches die Möglichkeit zur Simulation des Transportsystems bietet und bidirektional Daten mit Komponenten des realen Systems austauschen kann (vgl. Kuehner et al. 2021).

Im Kontext des Forschungsprojektes „Digitale Zwillinge zur dynamischen Simulation für die Planung und Steuerung innerbetrieblicher Transportsysteme im Rahmen der digitalen Fabrik“ (TwInTraSys) wird ein Digitaler Zwilling für innerbetriebliche Transportsysteme entwickelt und getestet. Der Digitale Zwilling soll operative Entscheidungen unterstützen, beispielsweise die Ressourceneinsatzplanung oder die Einstellung von Entscheidungsparametern für ein Transportleitsystem (TLS). Eine Zielsetzung im Projekt ist die automatisierte Erstellung von Simulationsmodellen, die den aktuellen Systemzustand abbilden. Für eine automatisierte Modellgenerierung müssen Informationen zum aktuellen Wegenetz (Knoten und Kanten) und zum Prozess übertragen werden. Für die Abbildung des realen Systemverhaltens in der Simulation, ist das Entscheidungsverhalten der steuernden Instanz (TLS) zu synchronisieren. Um den Implementierungsaufwand zu reduzieren, soll das reale TLS auch die Entscheidungen innerhalb des virtuellen Systems (Simulation) treffen. Damit wird sichergestellt, dass Entscheidungen im realen und im virtuellen System nach denselben Regeln getroffen werden.

2 Wissenschaftliche Abgrenzung

In der Literatur finden sich zahlreiche Ansätze zur automatischen Modellgenerierung (AMG) für die Simulation. Eine Klassifizierung der Vorgehensweise wird von Bergmann und Straßburger (2020) vorgestellt. Reinhardt et al. (2019) geben einen Überblick über die genutzte Datengrundlage und das eingesetzte Datenaustauschformat. In zahlreichen Veröffentlichungen wird AutomationML als Datenaustauschformat für Produktions- und Logistiksysteme vorgeschlagen. (Schroeder et al. 2016, Prior et al. 2023) Im Kontext der AMG wird immer häufiger das Thema Virtuelle Inbetriebnahme (VIBN) diskutiert, mit dem Ziel den Vorbereitungsaufwand für die VIBN zu reduzieren. (Prior et al. 2023) Die VIBN dient laut VDI 3693 zur Aufdeckung von Fehlern aus dem Engineering und findet vor der realen Nutzung des technischen Systems statt. Die Emulation ist dabei die funktionale Nachbildung des Systems durch ein Ersatzsystem. (VDI 3693) Das Ersatzsystem wird im Bereich der Intralogistik häufig mit Hilfe eines ereignisdiskreten Simulationsmodells umgesetzt. (Gutenschwager et al. 2000) Beispielsweise zeigen Kemper und Spiekermann (2010) den Status der Emulation von Logistiksystemen im SAP-Umfeld auf. Für die Kommunikation zwischen der Emulation und der getesteten Steuerung wird in Veröffentlichungen der OPC UA Standard für den Datenaustausch vorgeschlagen. (Grzona et al. 2019) Schmidl et al. (2021) demonstrieren, dass mit OPC UA auch der Datenaustausch zwischen verschiedenen IT-Systemen und Simulationsinstanzen möglich ist, was im Kontext von Co-Simulationen relevant ist. Im Kontext der Digitalen Zwillinge wird der Austausch von Daten zwischen betrieblichen Informationssystemen und dem vir-

tuellen Abbild wichtig. Koenig (2009) stellt dazu eine Möglichkeit vor, wie Lagerbestände aus dem SAP mit Hilfe von Remote Function Call (RFC) übertragen werden können. Die Übernahme von Fertigungsaufträgen mit Hilfe von RFC wird von Höpfe (2013) demonstriert. Die Möglichkeit zur Einbindung von Live-Daten in einen Digitalen Zwilling, für die Visualisierung von Informationen, stellt Selmaier et al. (2021) vor. Dabei wird eine Representational State Transfer Application Programming Interface (REST-API) eingesetzt.

Der vorliegende Beitrag greift bestehende Ansätze aus der Literatur auf und überträgt diese auf den im Kapitel 1 beschriebenen Anwendungsfall. Erstmals wird dabei die technische Umsetzung der Kommunikation zwischen einem Simulationsframework und dem ERP-System SAP über die Business Technology Platform (Cloudumgebung, BTP) beschrieben. Weiterhin wird dargestellt, wie Entscheidungssysteme aus betrieblichen IT-Systemen direkt in den Simulationsablauf eingebunden werden können und welche Auswirkungen dies auf die Laufzeit der Simulation hat.

3 Kommunikation zwischen SAP und Simulation

Die Kommunikation zwischen der Simulation und dem ERP-System SAP erfolgt über die Cloud Plattform (BTP). Auf der Plattform sind verschiedene Services verfügbar, unter anderem ein TLS-Adapter der Firma Flexus, der https-Anfragen verarbeiten kann. Um die Datensicherheit zu gewährleisten, muss jede Anfrage an das TLS über einen OAuth2-Token authentifiziert werden. Durch Nutzung von SAP Anmeldedaten (SAP Legacy User der BTP) und den Client-Informationen des SAP-Systems kann aus dem Simulationsframework ein verschlüsselter Token mit temporärer Gültigkeit über die BTP erzeugt werden. Dieser wird lokal gespeichert und muss bei jeder Anfrage im Request-Header als Authentifizierung übergeben werden.

Anfragen, die aus der Simulation an den TLS-Adapter übergeben werden, werden über einen Cloud-Connector an das SAP-ERP System weitergeleitet. Entsprechend der Art der Anfrage erfolgt eine Bearbeitung im TLS-Core. Die hierfür benötigten Informationen aus der Simulationsumgebung sind in der Anfrage enthalten. Dazu wird für jede Anfrageart ein JavaScript Object Notation (JSON) definiert, welches im Request-Body der Anfrage übergeben wird. Gleiches trifft auch auf die Informationen zu, die aus dem TLS an die Simulation übermittelt werden. Der Aufbau und die Kommunikationswege sind in der Abbildung 1 dargestellt.

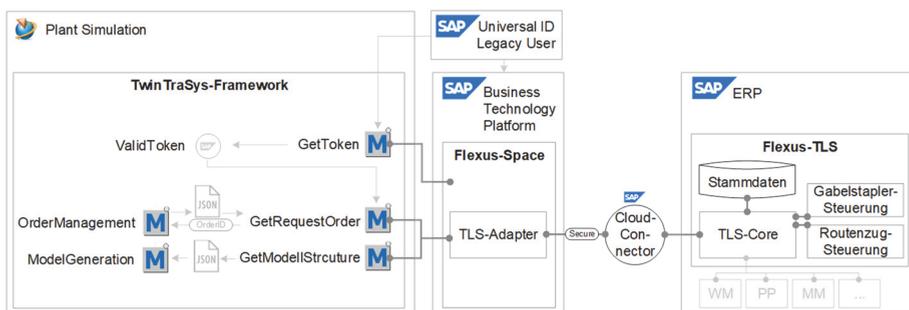


Abbildung 1: Darstellung der Kommunikation zwischen Simulation und TLS

Für die Kommunikation mit dem SAP wird im TwinTraSys-Framework (mit der Simulationssoftware „Plant Simulation“) ein eignes Netzwerk erstellt, welches die relevanten Informationen und Methoden für die Kommunikation zusammenfasst. Im Algorithmus 1 wird der vereinfachte Inhalt der Methode für einen PostRequest an den TLS-Adapter aufgezeigt. Zuerst wird das JSON-Objekt für den Header erzeugt (*reqHeader*). Dieser enthält alle notwendigen Informationen für die Authentifizierung. Zusätzlich wird ein weiteres JSON-Objekt als Request-Body erstellt, in dem die relevanten Informationen aus dem Simulationsmodell (bspw. die aktuell offenen Aufträge, *Reqbody*) für die Funktionsausführung im TLS aufbereitet werden. Für die Rückmeldung des TLS werden zwei zusätzliche JSON-Objekte definiert (*respheader*, *respbody*). Auf den Inhalt der JSON-Objekte und die Verarbeitung der Daten wird in den folgenden Kapiteln näher eingegangen.

Algorithmus 1: Auszug aus einer Methode für den PostRequest (SimTalk)

```
//Generierung Header
var reqHeader : json
reqHeader.getOrCreateJson("Authorization")
reqHeader["Authorization"] := ...
//Generierung Body
var strjs : string := Js_Request.asstring
var Reqbody : json
reqbody.parse(strjs)
//PostRequest an TLS-Adapter
var respheader : json
var respbody : json
httpPostRequest(Url, reqbody, reqheader, respheader, respbody)
//Verarbeitung des Response
...
```

Eine wichtige Anforderung bei der Gestaltung der Schnittstelle zwischen Simulation und SAP war, dass gleichzeitig mehrere Instanzen des Simulationsframeworks mit dem gleichen SAP-System kommunizieren können. Dies ist der Fall, wenn parallel mehrere Experimente durchgeführt werden sollen.

4 Generierung ablauffähiger Simulationsmodelle mithilfe von SAP-Stammdaten

Moderne Transportleitsysteme (TLS) verfügen über umfangreiche Stammdaten, mit denen das innerbetriebliche Transportsystem beschrieben wird. Dies ist notwendig, da das TLS bei der Auftragszuweisung bspw. das Wegenetz und die Einsatzbereiche von Flurförderzeugen berücksichtigen muss. Diese Stammdaten können für die Generierung von ablauffähigen Simulationsmodellen genutzt werden. Damit kann der Aufwand bei der Erstellung der Modelle reduziert sowie das Modell bei Veränderungen im realen System angepasst werden, was bei Digitalen Zwillingen unerlässlich ist. Der Ablauf der automatischen Modellgenerierung orientiert sich an dem von Bergmann und Straßburger (2020) vorgestellten internen Ansatz, bei dem die Algorithmen für die Generierung im Simulationsframework enthalten sind.

Das TLS der Firma Flexus bietet ein graphisches User Interface (*RouteOptimizer*) an, mit der das innerbetriebliche Transportsystem beschrieben werden kann (siehe Abbildung 3, links). Dabei wird das Wegenetz mit Hilfe von Knoten und Kanten definiert. Knoten können Verzweigungspunkte (Kreuzungen) oder Funktionsbereiche (Bereitstellfläche, Verladerrampen) sein. Zu jedem Knoten können Informationen hinterlegt werden, dies kann z.B. die Bezeichnung eines Produktversorgungsbereiches sein, mit dem ein Transportauftrag aus dem SAP diesem Knoten zugeordnet werden kann („positions“). Die Knoten werden im RouteOptimizer mit Wegen verbunden. Auch diese Wege können mit Informationen angereichert werden. Beispielsweise kann definiert werden, ob der Weg in beide Richtungen befahren werden kann („directed“) und welche Transportressourcen („vehicleTypes“) den Weg befahren dürfen. Aus diesen Informationen wird ein JSON-Objekt erstellt, welches sich inhaltlich in die Bereiche Knoten („nodes“), Kanten („edges“) und Fahrzeugtypen („vehicleTypes“) untergliedern lässt. Exemplarisch ist in der Abbildung 2 ein Auszug dargestellt.

```

"message": {      }, //Allgemeine Informationen
"area": {        //Es kann mehrere Bereiche geben (bspw. Stockwerke, Hallen)
  "nodes": [
    { "id": 1,           //ID des Knoten im TLS
      "x": 238,         //X-Position im Area-Koordinatensystem
      "y": 595,         //Y-Position im Area-Koordinatensystem
      "positions": [{}], //Angaben zum Lagerplatz bei dem ein Knoten befindet
      "externalNodes": [{}], ... //Angaben zu Verbindungen zu anderen Areas
    }
  ],
  "edges": [
    { "sourceNode": 1, //Start-Knoten für den Weg
      "destinationNode": 30, //Ziel-Knoten für den Weg
      "vehicleTypes": [...], //Fahrzeugtypen die den Weg befahren dürfen
      "directed": false, //zulässige Fahrtrichtung des Wegs
      "blocked": false }, ... //Wegabschnitt aktuell gesperrt
    }
  ],
  "vehicleTypes": [
    { "area": "", //Einsatzbereich des Fahrzeuges
      "vehicletype": "RTZ-A", //Bezeichnung des Fahrzeugtyps
      "speed": 3.5, //Geschwindigkeit des Fahrzeugtyps
      "maximumWeight": 300, //max. Gewicht einer Transporteinheit
      "vehicles": [{"vehicle": "RTZ1-A", ...} ], ... // Instanzen des Typs
    }
  ]
}

```

Abbildung 2: Auszug aus dem JSON-Objekt für die Modellgenerierung

Nach dem das JSON-Objekt an die Simulation übermittelt wurde, werden die Informationen für die Modellgenerierung aufbereitet. Dies umfasst eine Konsistenzprüfung der Daten und eine Aufteilung dieser auf verschiedene Tabellen. Das Simulationsframework umfasst verschiedene generische Bausteine mit denen innerbetriebliche Transportsysteme abgebildet werden können. Essenziell sind dabei Wegabschnitte, Kreuzungen und Übergabepunkte. Im ersten Schritt werden die Knoten entsprechend der Koordinaten erzeugt. Als nächstes werden die Kanten positioniert und mit den Knoten verbunden. Die Länge eines Wegeabschnittes wird dabei aus den Koordinaten der Start- und Endknoten berechnet. Bei der Erzeugung der Objekte in der Simulation erfolgt eine Parametrierung der Bausteine, so wird zum Beispiel die maximal zulässige Geschwindigkeit auf einem Wegstück in den Eigenschaften des Objektes „Weg“ gespeichert. Im letzten Schritt der Modellgenerierung werden die Transportressourcen auf Grundlage der Fahrzeugtypen erzeugt und soweit nicht anders spezifiziert zufällig

im Wegenetz platziert. Der Ablauf der Modellgenerierung ist in der Abbildung 3 graphisch dargestellt.

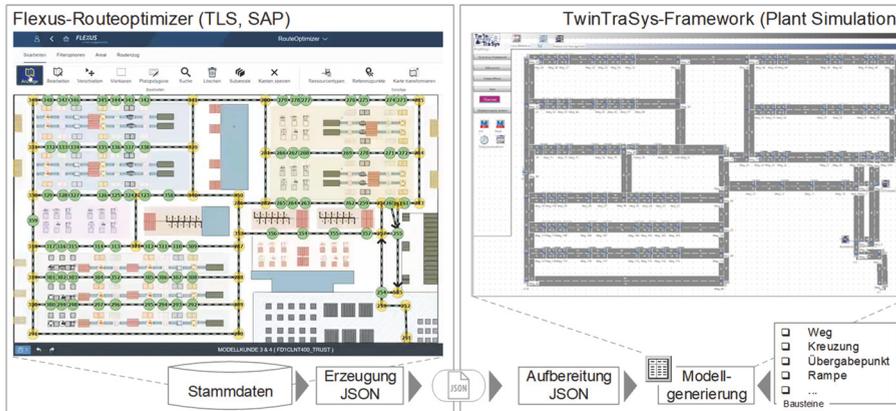


Abbildung 3: Ablauf der automatischen Modellgenerierung

5 Integration eines Transportleitsystems als Entscheidungsinstanz innerhalb der Simulation

Die elementare Funktion eines Transportleitsystems (TLS) ist die zielorientierte Zuweisung von Transportaufträgen für die einzelnen Transportressourcen. Dabei können Zielsetzungen wie die Reduzierung der Fahrstrecke, der Auftragsdurchlaufzeiten und des Leerfahrtanteils verfolgt werden. Für die Entscheidungsfindung beurteilt das TLS die im aktuellen Systemzustand verfügbaren Transportaufträge mit Hilfe einer Zielfunktion, die projektspezifische Gewichtungen der einzelnen Ziele berücksichtigt.

Für die realitätsnahe Abbildung des Systemverhaltens im Simulationsframework muss die Entscheidungslogik des TLS in der Simulation implementiert oder das TLS in den Simulationsablauf einbezogen werden. Um die Synchronisation zwischen realem und virtuellem System zu unterstützen, wird die Einbindung des SAP-TLS in den Simulationsablauf umgesetzt.

```
"singleOrderOptions": {
  "orders": [ //Menge der offenen Aufträge mit folgenden Informationen
    {"orderId": "0000100921 ",
      "nodes":[
        {"nodeId": "AP3-3",
          "sequenceId": 0,
          "released": true,
          "nodePosition": {"x":10,"y":20}}, ... ]
      "createdAt": "2023-03-30T11:40:03.12Z"},...],
    "vehicles":[ //Liste an einsetzbaren Fahrzeugen
      {"vehicleName": " RTZ1-A",
        "nodeId": "12"},...],
    "prioritySettings": //List mit experimentspezifischen Parametern
      {"distance":[
        {"upToMeters": 50,
          "value": 100},...],
        "time": [...],...}
```

Abbildung 4: Auszug aus dem JSON-Objekt für die Auftragszuweisung

Im vorgestellten Ansatz werden alle Transportaufträge in der Simulation verwaltet, damit das SAP-System gleichzeitig von mehreren parallellaufenden Simulationen genutzt werden kann. Weiterhin sind alle experimentspezifischen Parameter, die das TLS betreffen, auch in der entsprechenden Instanz der Simulation gespeichert. Dies sind bspw. Gewichtungsfaktoren der Zielfunktion, da sich diese in den Experimenten unterscheiden können.

Ausgangspunkt ist der Abschluss eines Transportauftrages durch eine Transportressource. In diesem Fall ruft die Transportressource den Auftragsmanager in der Simulation auf. Die Transportauftragssteuerung erstellt eine Liste mit aktuell verfügbaren Transportaufträgen für die entsprechende Transportressource. Durch die Aufbereitung der Aufträge kann die Datenmenge, die zwischen Simulation und SAP ausgetauscht wird, reduziert werden. Anschließend wird ein JSON-Objekt erstellt, welches die aufbereiteten Transportaufträge („*orders*“), Angaben zu den aktuell verfügbaren Fahrzeugen („*vehicles*“) und die experimentspezifischen TLS-Parameter („*prioritySettings*“) enthält. Die Struktur des JSON orientiert sich hierbei an der in der VDA5050 (2022) beschriebenen Auftragsstruktur für die Kommunikation zwischen FTF und der Leitsteuerung. Das entsprechende JSON-Objekt ist in Auszügen in der Abbildung 4 dargestellt.

Die übermittelten Daten werden im TLS-Core (SAP) aufbereitet und an die Funktionen für die Auftragszuweisung übergeben. Bei der Implementierung dieser Funktionen im SAP ist eine Trennung der logischen Funktionen (Auswahl des nächsten Auftrages) und der Datenverarbeitung (Eintragungen in die SAP-Datenbank) notwendig. Damit wird vermieden, dass ein Aufruf des TLS aus der Simulation zu einer Veränderung von Daten im realen System führt. Das Ergebnis der Auftragszuweisung wird in ein JSON-Objekt überführt und an die Simulation übergeben. In der Simulation wird der Auftrag an die entsprechende Transportressource weitergegeben und die Daten in der Transportauftragstabelle werden aktualisiert. Der schematische Ablauf ist in der Abbildung 5 visualisiert.

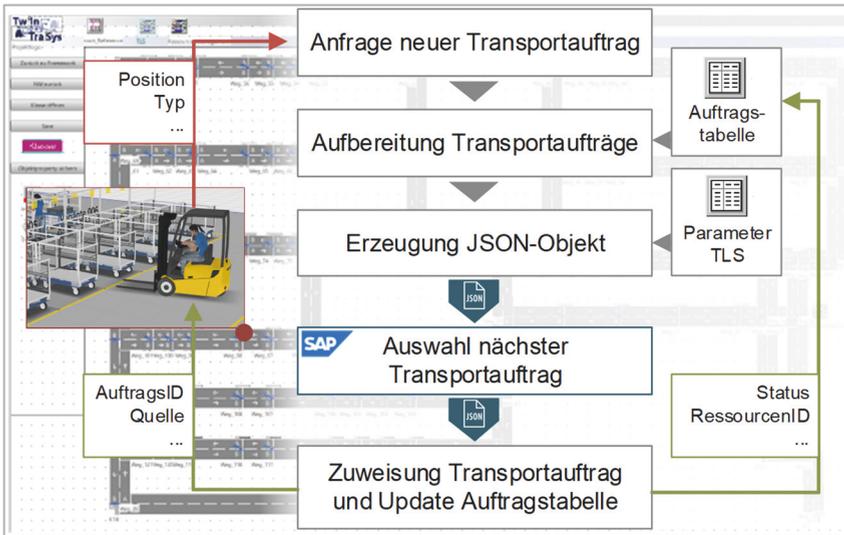


Abbildung 5: Ablauf bei der Zuweisung von Transportaufträgen durch das SAP-TLS

Falls in dem innerbetrieblichen Transportsystem Routenzüge zum Einsatz kommen, wird die Tourenbildung (Zusammenfassen von Aufträgen zu einer Fahrt des Routenzuges) durch das TLS übernommen. In diesem Fall können auch mehrere Transportaufträge an eine Ressource übermittelt werden. Gleiches ist auch möglich, wenn die Transportkapazität des Gabelstaplers größer als eine Transporteinheit ist.

Für den dargestellten Anwendungsfall ist es wichtig, dass die Laufzeit der Simulation in einem akzeptablen Rahmen liegt, da für die Absicherung von Entscheidungen mehrere Experimente und Replikationen durchgeführt werden müssen. Während die Modellgenerierung nur ausgeführt wird, wenn sich Änderungen im System ergeben haben, erfolgt die Zuweisung der Transportaufträge deutlich häufiger (>1.000 Zuweisungen pro simulierten Tag). Im Rahmen eines Massentests, bei dem innerhalb eines kurzen Zeitraums viele Anfragen an das SAP-TLS übertragen wurden, wurde die Laufzeit des Simulationsmodells untersucht. Bei dieser Untersuchung wurde die Anzahl, der an das TLS, übergebenen Transportaufträge variiert, damit der Einfluss dieses Kriteriums berücksichtigt werden kann. Hierfür wurde die Simulation solange angehalten, bis das TLS den nächsten Transportauftrag zurückgeliefert hat. Die Ergebnisse des Tests sind in Tabelle 1 aufgeführt.

Tabelle 1: Laufzeiten in Abhängigkeit der Anfragengröße

Anzahl Aufträge pro Request	Zeit pro Request	Laufzeit der Simulation bei 1.000 Request
1	0,059 s	1,04 min
3	0,184 s	3,13 min
5	0,300 s	5,05 min
8	0,445 s	7,50 min
10	0,550 s	9,28 min
13	0,692 s	11,63 min
15	0,793 s	13,42 min

Die Ergebnisse zeigen, dass die Zeit für die Kommunikation und Entscheidungsfindung im TLS bei einer Erhöhung der Anzahl an möglichen Aufträgen (Auftragsaufbereitung in der Simulation), die an das TLS übergeben wird, linear steigt. Daraus lässt sich die Schlussfolgerung ableiten, dass eine gezielte Auswahl an Transportaufträgen sich positiv auf die Laufzeit der Simulation auswirkt. Dabei ist sicherzustellen, dass das Systemverhalten nicht beeinflusst wird, respektiv die Entscheidung des TLS verändert wird.

6 Herausforderungen bei der Integration eines realen Transportleitsystems

Die Integration eines realen Transportleitsystems (TLS) in eine Simulationsumgebung hat den Vorteil, dass dessen Entscheidungslogik nicht in der Simulation nachgebildet werden muss. Weiterhin ziehen Anpassungen im TLS keine Änderungen in der Simulation nach sich, was die Nutzbarkeit und Aktualität des Modells verbessert. Die Umsetzung im Forschungsprojekt TwinTraSys hat aufgezeigt, dass die Integration Herausforderungen mit sich bringt.

Im SAP werden unternehmensrelevante Informationen gespeichert und verarbeitet. Diese Informationen müssen vor Fremdzugriff und ungewollter Modifikation geschützt werden. Aus diesem Grund muss die Kommunikation zwischen SAP und Simulation die Anforderungen an die Datensicherheit erfüllen. Damit geht auch die Forderung einher, dass Simulationsereignisse nicht die Daten im operativ genutzten SAP-System verändern dürfen. Unter Berücksichtigung der Anforderung, dass gleichzeitig mehrere Simulationsmodelle mit dem gleichen SAP-System kommunizieren müssen, führt dies zu der Notwendigkeit, dass die Datenspeicherung im Simulationsframework (Modell oder Datenbank) erfolgen muss. Dies hat zur Folge, dass die Datenmenge, die bei einer Anfrage an das TLS übermittelt werden muss, zunimmt. Weiterhin hat sich gezeigt, dass eine klare Trennung der Funktionen im SAP sinnvoll ist. Häufig sind die Entscheidungslogiken eng mit der Ausführung von Transaktionen auf der SAP-Datenbank verbunden. Ohne diese Trennung besteht die Gefahr, dass Inkonsistenzen entstehen und das reale System beeinträchtigt wird.

Im realen Einsatz ist die Frequenz für Anfragen an das TLS gering, da die Transportressourcen die Aufträge zuerst ausführen müssen, bevor eine neue Anfrage gestellt wird. Die Frequenz erhöht sich deutlich, wenn die Anfragen durch eine oder mehrere

Simulationsinstanzen generiert werden. In diesem Fall muss das Lastmanagement des SAP-Servers beachtet werden, damit der Digitale Zwilling nicht das reale System ausbremst.

Es ist anzumerken, dass durch die Funktionen für die Kommunikation mit einem http[s]-Server von Plant Simulation eine Anbindung der Simulation mit dem BTP gut umsetzbar ist. Hierbei gestaltet sich jedoch die Aufbereitung und Erstellung von komplexeren JSON-strukturen für die Anfragen vergleichsweise aufwändig.

Wie sich im Massentest gezeigt hat, treten lange Simulationslaufzeiten durch die Kommunikation mit dem Transportleitsystem auf. In Hinblick auf die Menge an Experimenten, die im Forschungsprojekt TwInTraSys durchgeführt werden, soll daher im weiteren Projektverlauf die Schnittstelle hinsichtlich der Laufzeit optimiert werden.

Danksagung

Diese Arbeit wurde im Rahmen des Forschungsprojekts „Digitale Zwillinge zur dynamischen Simulation für die Planung und Steuerung innerbetrieblicher Transportsysteme im Rahmen der digitalen Fabrik“ (TwInTraSys) durchgeführt. Das Projekt wird durch das Bayerische Staatsministerium für Wirtschaft, Landesentwicklung und Energie im Rahmen des Bayerischen Verbundförderprogramms – Förderlinie Digitalisierung Förderbereich Informations- und Kommunikationstechnik gefördert (Förderkennzeichen DIK-2104-0062).

Literaturverzeichnis

- Bergmann, S.; Straßburger, S.: Automatische Modellgenerierung – Stand, Klassifizierung und ein Anwendungsbeispiel. In: Mayer, G.; Pöge, C.; Spieckermann, S.; Wenzel, S. (Hrsg.): Ablaufsimulation in der Automobilindustrie. Springer Berlin Heidelberg 2020, S. 333–347.
- Fottner, J., Galka, S., Habenicht, S., Klenk, E., Meinhardt, I., Schmidt, T. Planung von innerbetrieblichen Transportsystemen. Springer, Berlin, Heidelberg 2022.
- Grzona, P., Knüpfer, P., Hörmann, R., Wilsky, P., Horler, S. Anwendungsorientierte Simulationsbausteine für die schlanke virtuelle Inbetriebnahme verketteter Produktionssysteme. Putz, M, Schlegel, A. (Hrsg.): Simulation in Produktion und Logistik 2019. Auerbach, 2019, S. 562-569.
- Gutenschwager, K., Fauth, K., Spieckermann, S., Voß, S. Qualitätssicherung lagerlogistischer Steuerungssoftware durch Simulation. Informatik-Spektrum 23, 2000, S. 26-37.
- Höppe, N., Seeaner, F., Spieckermann, S. Simulationsgestützte Feinplanung im Produktionsumfeld - vom SAP zur SPS. In Dangelmaier, W., Laroque, C., Klaas, A. (Hrsg.) Simulation in Produktion und Logistik Entscheidungsunterstützung von der Planung bis zur Steuerung, Paderborn, 2013, S. 669-678.
- Kauke, D., Galka, S., Fottner, J. Digital Twins in Order Picking Systems for Operational Decision Support. In 54th Hawaii International Conference on System Sciences, 2001, S. 1655–1664.
- Kemper, J., Spieckermann, S. Emulation von Logistik-Steuerungen in SAP-Umgebungen. In Zülch, G., Stock, P. (Hrsg.): Integrationsaspekte der Simulation: Technik, Organisation und Personal. Karlsruhe, 2010, S 583-590.

- Koenig, J. Das Virtual Engineering Interface: VEI 4 SAP - Die Kopplung von SAP und Plant Simulation. https://media1.autohaus.de/fm/3576/Snapshots_SAP_Simulation_Schnittstelle.pdf, 2009, letzter Zugriff 15.05.2023.
- Kuehner, K.J., Scheer, R., Straßburger, S.: Digital Twin: Finding Common Ground – A Meta-Review, IN: Mourtzis, D. (Hrsg.): *Procedia CIRP*, Volume 104, 2021, S. 1227-1232.
- Prior, J., Karch, S., Strahilov, A., Kuhlenkötter, B., Lüder, A. Template-Based Production Modules in Plant Engineering. In Liewald, M., Verl, A., Bauernhansl, T., Möhring, HC. (Hrsg.): *Production at the Leading Edge of Technology..* Springer, Cham, 2023, S. 652-663.
- Reinhardt, H., Weber, M., Putz, M. A Survey on Automatic Model Generation for Material Flow Simulation in Discrete Manufacturing. *Procedia CIRP*. Volume 81, 2019, S 121-126.
- Schmidl, E., Wenk, M., Russwurm, E., Franke, J. Interoperabilität realer und simulierter Produktionssysteme mittels OPC UA. In Franke, J. Schuderer, P. (Hrsg.): *Simulation in Produktion und Logistik 2021*. Cuvillier Verlag, Göttingen, 2021, S. 496-504.
- Schroeder, G.N., Steinmetz, C., Pereira, C.E., Espindola, D.B. Digital twin data modeling with automationml and a communication methodology for data exchange. *IFAC-PapersOnLine*, 49(30), 2016, S.12-17.
- Selmaier, A., Sjarov, M., Herbert, M., Franke, J., Distler, J., Fürst, J. Der Digitale Zwilling als echtzeitnahes Fertigungsabbild. In Franke, J. Schuderer, P. (Hrsg.): *Simulation in Produktion und Logistik 2021*. Cuvillier Verlag, Göttingen, 2021, S. 496-504.
- VDA 5050. Schnittstelle zur Kommunikation zwischen Fahrerlosen Transportfahrzeugen (FTF) und einer Leitsteuerung. Version 2.0, 2022.
- VDI/VDE 3693. Virtuelle Inbetriebnahme Modellarten und Glossar. Beuth, Berlin.